YOUR OFFICIAL RASPBERRY PI MAGAZINE

The MagPi

The official Raspberry Pi magazine

Issue 52 December 2016

raspberrypi.org/magpi

MERRY CHRISTMAS MAKES

Get creative with four festive projects

AMIGA POWER!

Turn your Pi into the ultimate emulation station

BEGINNER ELECTRONICS

Get started with prototyping on your Pi

BUILD A DIY ALARM CLOCK

Wake at dawn with a sunrise alarm

GIVE YOUR ROBOT BRAINS

How to add a wealth of sensors to your Pi robot

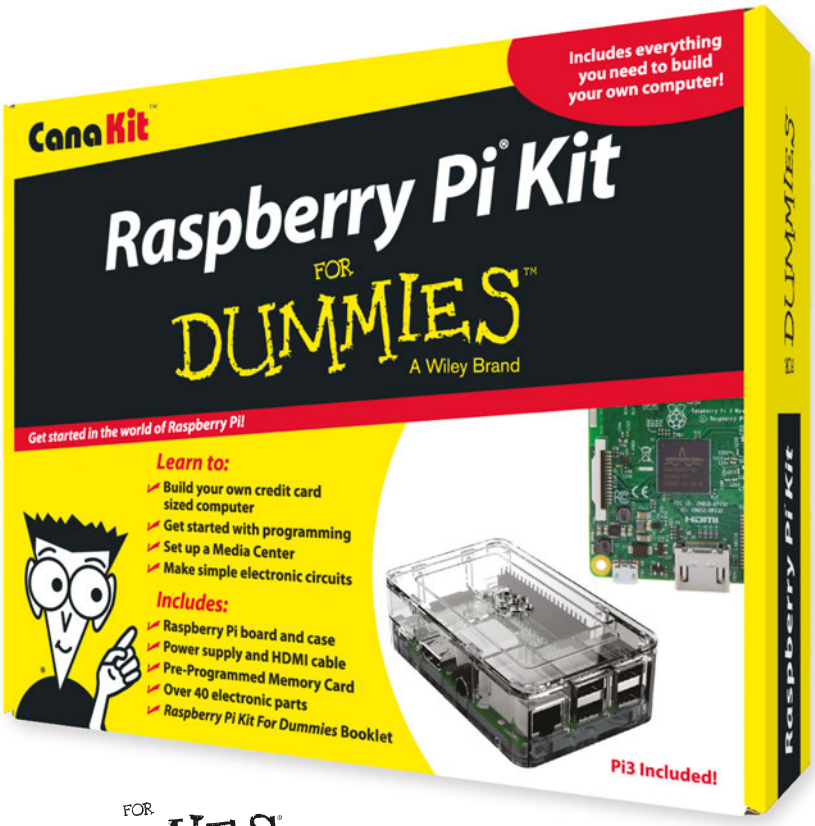
Issue 52 • Dec 2016 • £5.99

9 772051 998001 12

THE ONLY PI MAGAZINE WRITTEN BY THE RASPBERRY PI COMMUNITY

CanaKit™

The Perfect Holiday Gift!



FOR DUMMIES® A Wiley Brand

Available for worldwide shipping at:

WWW.CANAKIT.COM

Available in Europe through RS Components

RS

Kit Includes:

- *Raspberry Pi For Dummies* Booklet
- Raspberry Pi 3 Board
- Memory Card
- Plastic Case
- 2.5A Power Supply
- HDMI Cable
- Resistors
- LEDs
- Push Button Switches
- Prototyping Breadboard
- Jumper Wires
- Heat Sinks

$89.99 US DOLLARS

£69.99 EXCLUDING VAT

Raspberry Pi is a registered trademark of the Raspberry Pi Foundation. For Dummies and the Dummies Man logo are trademarks or registered trademarks of John Wiley & Sons, Inc. Used under license. RS logo is a registered trademark of RS Components Ltd. CanaKit is a registered trademark of Cana Kit Corporation.

ELECTRONIC KITS • ELECTRONIC PARTS • RASPBERRY PI • ARDUINO

WELCOME TO THE OFFICIAL MAGAZINE

The holiday season is upon us and that can mean lots of different things for lots of different people. For me it means spending some quality time with my family and keeping my kids entertained with games and fun projects. We've done our best to come up with some project ideas in our Christmas issue, including a great way to turn our Christmas tree cover into a blinky light sensation. Grab some LEDs, a pair of scissors, and turn to page 14 to get started.

If you're new to electronics, we've got you covered in our Raspberry Pi 101 tutorials starting on page 40. It's easier (and much more fun) than you might think.

If you're a bit old-school like me, you might be interested in reliving the glory days of the Amiga. Check out our expert guide starting on page 58 to learn how to emulate one of the greatest computers ever made on one of the greatest computers ever made!

Enjoy the issue.

Russell Barnes
Managing Editor

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
MERRY CHRISTMAS MAKES
AMIGA POWER!
Subscribe by 7th DEC 2016
to get your FREE PI ZERO in time for Christmas!
SUBSCRIBE & SAVE 25%
SEE PAGE 28 FOR DETAILS

THIS MONTH:

14 MERRY CHRISTMAS MAKES
Add some shiny blinking lights to this month's cover!

40 GET STARTED WITH CIRCUITS
Top-notch beginner advice for breadboards and GPIO Zero

58 RELIVE THE GLORY DAYS
Emulate the Amiga, one of greatest computers ever made

68 GIVE YOUR ROBOT BRAINS
Build on last month's amazing robot to add sensors and smarts

FIND US ONLINE raspberrypi.org/magpi **GET IN TOUCH** magpi@raspberrypi.org

The MagPi

Available on the App Store

GET IT ON Google Play

CC BY NC SA

EDITORIAL
Managing Editor: **Russell Barnes** russell@raspberrypi.org
Features Editor: **Rob Zwetsloot**
News Editor: **Lucy Hattersley**
Sub Editors: **Laura Clay, Phil King, Lorna Lynch**

DESIGN
Critical Media: **criticalmedia.co.uk**
Head of Design: **Dougal Matthews**
Designers: **Lee Allen, Mike Kay**

PUBLISHING
For advertising & licensing: **russell@raspberrypi.org +44 (0)7904 766523**
Publisher: **Liz Upton**
CEO: **Eben Upton**

DISTRIBUTION
Seymour Distribution Ltd
2 East Poultry Ave
London
EC1A 9PT | **+44 (0)207 429 4000**

SUBSCRIPTIONS
Select Publisher Services Ltd
PO Box 6337
Bournemouth
BH1 9EH | **+44 (0)1202 586 848**

CONTRIBUTORS
Alex Bate, Henry Budden, Brian Corteil, Mike Cook, Gareth Halfacree, James Hobro, Richard Hayler, Phil King, Simon Long, K.G. Orphanides, William Plappert, Matt Richardson &Richard Smedley

recycle When you have finished with this magazine, please recycle it.

This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., 30 Station Road, Cambridge, CB1 2JH. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2051-9982.

Contents

Issue 52 Christmas 2016 raspberrypi.org/magpi

TUTORIALS

> PI 101 – USE BREADBOARDS **40**
Learn how to prototype circuits on a Raspberry Pi

> PI 101 – MASTER GPIO ZERO **42**
Program your circuits with the powerful GPIO Zero

> PI 101 – USE VNC VIEWER **46**
Remotely connect to your Pi's desktop

> INTRODUCTION TO C PART 6 **48**
This month, split your code up into functions

> MAKE A CHRISTMAS GAME **50**
Create a game of sliders in the Pi Bakery

> BUILD A SUNRISE ALARM CLOCK **56**
Get up at dawn with this Pi-powered alarm

> AMIGA EMULATION EXPLAINED **58**
Get your retro gaming on by emulating this classic

> PARALLEL PROGRAMMING **60**
Perform multiple calculations at the same time

> MULTI-BOOT YOUR PI PART 1 **62**
Start building a multi-booting Raspberry Pi

IN THE NEWS

ASTRO PI AWARD

Pi's own Dave Honess wins the Arthur Clarke award for his work on Astro Pi!

12

COVER FEATURE

14

MAKER XMAS

THE PIONEERS

Raspberry Pi's new initiative to get teens into coding

6

EDUCATOR'S MAGPI

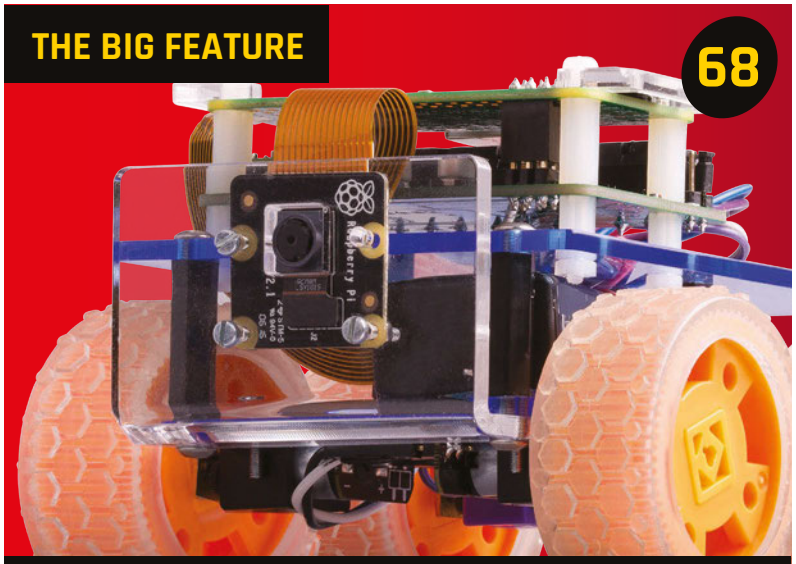
FREE RESOURCES FOR THE CLASSROOM
The MagPi
EDUCATOR'S EDITION
TEACH AND MAKE WITH RASPBERRY PI
FROM THE MAKERS OF THE OFFICIAL RASPBERRY PI MAGAZINE

A new MagPi Educator's Edition is out now

11

Contents

THE BIG FEATURE



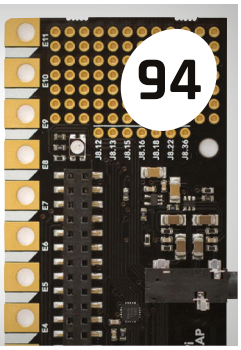
68

ROBOT SENSOR CHALLENGES

Add sensors to our robot from last issue to make it able to compete in Pi Wars challenges

WIN

A PI CAP AND A TUBE OF ELECTRIC PAINT



94

SPY V. SPI

Young secret agents learn to code in this fun game

86

YOUR PROJECTS

30

SISYPHUS

One part table, one part work of art

IOT DOOR 34

A facial recognition door using Windows 10

PI LOOM 36

Weaving has never been easier

REGULARS

> NEWS 06

> TECHNICAL FAQ 64

> BOOK REVIEWS 82

> THE FINAL WORD 96

COMMUNITY

> THIS MONTH IN PI 84
What else happened this month in the world of Pi?

> COMMUNITY SPOTLIGHT 88
We talk to Alex Eames, the man behind RasPi.TV

> EVENTS 90
Find a community event near you

> LETTERS 92
We answer your letters about the magazine and Pi

REVIEWS

> PI CAP 76

> DREMEL 78

> SERVO KIT 80

> RASPBERRY SWEETS 81

THE PIONEERS

New Raspberry Pi programme for teens needs mentors and clubs to help out

#MAKEYOURIDEAS

Above A series of competitions will inspire young makers to turn their ideas into reality

The Raspberry Pi Foundation needs your help to keep teenagers making cool stuff with code and technology.

The Pioneers project is going to be a new programme for coding clubs and teen makers. It will launch in the new year, and it will need both enthusiastic teens and adult volunteers.

JOIN PIONEERS

The Pioneers challenge is open to everyone in the UK between the ages of 12 and 15. It's for teams of four or under. The Raspberry Pi Foundation asks that each team finds someone over the age of 18 to act as a mentor.
raspberrypi.org/pioneers
#MakeYourIdea

Teenagers in the UK can become Pioneers by signing up online at **raspberrypi.org/pioneers**. The hashtag #MakeYourIdeas will be used to share projects.

"We want to find and support teenage digital makers in the UK," says Rob Buckland, director of programmes. "All over the world, teenagers are building cool stuff, learning how to bend digital technology to solve problems they care about, and having lots of fun."

More importantly, says Rob, "they are the next generation of inventors, entrepreneurs, and makers. We want to support them. The aim of Pioneers is to provide guidance, inspiration, and mentorship to teenage makers, and to the adults who mentor them."

Every school term, Raspberry Pi will set a new mission for the Pioneers community. Each of these challenges will have a different theme.

"They will win cool swag and money-can't-buy experiences," Rob reveals.

For young makers "it's also a chance to work with a team of like-minded people to create an idea they love," says Rob. "Something that is relevant to them.

"Importantly, it's also a chance for teenagers to show the Raspberry Pi Foundation and maker community what they can do," he explains.

So teenage Pioneers get to win both prizes and kudos. It sounds like a great idea to us.

The first Pioneers competition will launch in January 2017. The first challenge will be announced early next year, and we can't wait to hear what it is.

New recruits

Pioneers will be aged between 12 and 15. "Get together with some buddies and form a team," suggests Rob. Up to four Pioneers can be in a team.

"There is no right or wrong way to start a Pioneers team," says Olympia Brown, the senior programme manager who will be running the Pioneers scheme. "It can be motivated by students or inspired by a mentor."

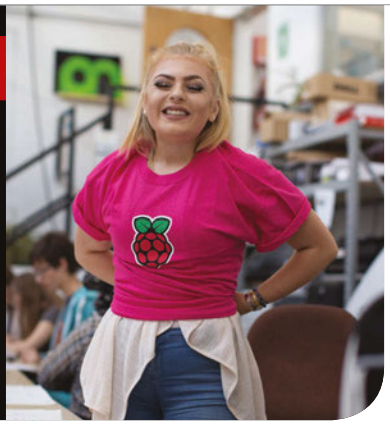
There is one condition, according to Olympia: "We just ask that each team finds someone over the age of 18 to act as a mentor."

Pioneers will start their first mission in January 2017. "Each team has to produce a video of their work to show the judges and the rest of the world," says Olympia.

Projects will be judged, and the best ones will win prizes. "We all like to be winners," says Rob, "but it's a great chance to get together with like-minded, creative souls and start a new community…

BECOME A MENTOR

Mentors can be any responsible adult who wishes to guide teenagers on the project. They may run a makerspace or CoderDojo, or be a school teacher, scout leader, or parent. There is no right or wrong way to build a team. Taking part can be motivated by students or inspired by a mentor. raspberrypi.org/pioneers



A community where we share skills, make our ideas a reality, occasionally blow things up, and lead the way for the future in an increasingly digital world."

Mentoring

Pioneers will need guidance and help, and as such the Raspberry Pi Foundation is looking for adult mentors to provide this. "It's a chance to help young people on their journey to expand their digital making skills," comments Olympia.

The Raspberry Pi Foundation hopes that mentors will come in all shapes and sizes. They could be already involved with the Raspberry Pi community. "We expect mentors to be running a CoderDojo, a makerspace, or a similar club," says Rob.

"We designed Pioneers so that existing computing clubs and CoderDojos can get involved," explains Philip Colligan, CEO of Raspberry Pi. "It's also a great excuse to get together for the first time and we're hoping that we will see many more young people getting creative with technology."

Pioneers will also provide a much-needed next step for teenagers who have outgrown Code Club. While Code Club is designed to introduce young makers to coding, Pioneers will enable teens to show what they can create.

"Pioneers will provide an exciting follow-on activity for children coming from Code Club to continue to develop their coding skills and be creative," states Maria Quevedo, director of Code Club UK. "It will be a great opportunity for them to continue making things with computers and to show off their work to others."

Giustina Mizzoni, executive director at CoderDojo, agrees. "There is so much creativity and talent in the community," she tells us. "One place where it is evident is in our annual Coolest Projects Awards. I was inspired by so many unique and interesting projects... Orla, 16, designed a mobile app Key Tracker, using Bluetooth, while Carl and Leo (both 14) built a Braille label puncher that converted text to Braille and printed it on thin aluminium plates."

"The Raspberry Pi Foundation is focused on putting digital making into the hands of everyone, and we are passionate about making sure there is provision for all ages," asserts Rob. "That's where Pioneers comes in."

"We're really excited to launch Pioneers," says Philip. "There's already an amazing community of young digital makers out there. We want to celebrate what they're achieving and challenge them to do even more."

Above Pioneers will challenge teenagers to create amazing things with technology

TEEN SPIRIT

Take inspiration from these award-winning projects

Flood Gauge

Shane Fahy was just 11 when he started CoderDojo Athenry in 2013. He received his Future Makers Award in the RDS Dublin on 18 June 2016 for his flood detector project. "Don't think of it as a competition – do it because you enjoy it," advises Shane.
magpi.cc/2g2KwLZ

Student Essentials

Jack Underwood, 12, won the Innovation Award from Benchmark Recruitment for his app, Student Essentials. The app helps students in the classroom and includes a Homework Tracker, Measurement Converter, Translator, Notepad, QR Code Scanner, and more.
magpi.cc/2g2OiF8

Sign Language Translator

Ryan Patterson was 17 when he built this Sign Language Translator. It was a Grand Award winner in the Intel International Science and Engineering Fair. The glove detects hand movements used in ASL and translates them into letters.
magpi.cc/2g2NiRD

PI FINANCE

Economists are testing out macroeconomic models using the Raspberry Pi computer

Bank Underground, the Bank of England's blog, reports that Raspberry Pi computers are being used to test out macroeconomic modelling systems.

"There was a time when macro-modelling was an expensive business," says Andrew Blake, senior advisor in monetary analysis at the Centre for Central Banking Studies. "You needed economic theorists, at least a couple of tame econometricians, someone who could actually program, research assistants to delve into the long-abandoned stacks of library basements to source data, and alchemists well-versed in the dark arts of forecasting and simulation.

"Nowadays: not so much," he tells us. "Many of the world's leading models are given away free via easily accessible working papers."

To show you just far we've come, Andrew built a modern macroeconomic model using the COMPASS economic model. This was built and tested to provide economic forecasting on a Raspberry Pi Zero computer.

> "There are no barriers to entering the macro-modelling world left"

"There are no barriers to entering the macro-modelling world left," concludes Andrew. "Not even the cost of the computer. You don't have to run it on a Raspberry Pi, but if you do, it is remarkably straightforward, ridiculously entertaining, and properly educational."

Full details of the project can be found on the Bank Underground blog (**magpi.cc/2fHsL2q**).

SPINACH BOMB DETECTOR

MIT engineers are using Raspberry Pi boards and spinach plants to detect explosives.

The remarkable new technology is called 'plant nanobionics'.

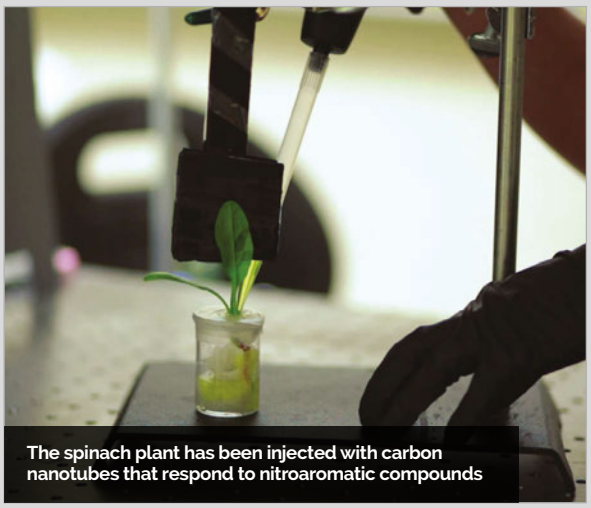
Spinach plants are injected with carbon nanotubes, and the leaves are then scanned by a Raspberry Pi and infrared camera.

"The plants were designed to detect chemical compounds known as nitroaromatics, which are often used in landmines and other explosives," says Anne Trafton from the MIT News Office.

"When one of these chemicals is present in the groundwater, which is sampled naturally by the plant, carbon nanotubes embedded in the plant leaves emit a fluorescent signal that can be read with an infrared camera."

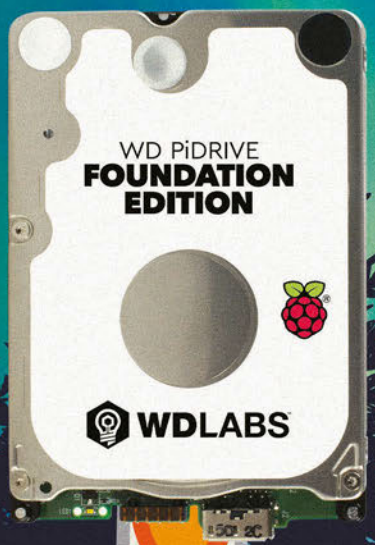
"This is a novel demonstration of how we have overcome the plant/human communication barrier," says Michael Strano, the Carbon P. Dubbs professor of chemical engineering at MIT.

A video of the remarkable project can be watched online at MIT (**magpi.cc/2fHxFfC**).



The spinach plant has been injected with carbon nanotubes that respond to nitroaromatic compounds

A CUSTOM GIFT FOR YOUR RASPBERRY PI



Combining SD card and USB drive functionality, the **WD PiDrive Foundation Edition** is preloaded with custom software to help you start creating Raspberry Pi projects quickly and easily.

WD PIDRIVE FOUNDATION EDITION

USB FLASH 64GB

$18.99

USB Flash Drive
SanDisk MicroSD™ card

WD PIDRIVE FOUNDATION EDITION

USB HARD DRIVE 250GB

$28.99

USB 2.5" 7mm HDD
SanDisk MicroSD™ card
WD PiDrive Cable

WD PIDRIVE FOUNDATION EDITION

USB HARD DRIVE 375GB

$37.49

USB 2.5" 7mm HDD
SanDisk MicroSD™ card
WD PiDrive Cable

WDLABS

AVAILABLE NOW!

wdlabs.wd.com/holiday

COMPUTER AID CONNECT

Raspberry Pi devices recycled into internet education routers

Above The Computer Aid Connect brings offline internet access to remote parts of the world

"Computer Aid brings access to digital technology to some of the world's poorest people. The charity's latest project uses recycled Raspberry Pis to provide offline classrooms with a wealth of educational resources.

Computer Aid Connect is a portable hotspot device based on the Raspberry Pi. "It is for offline students, and teachers, across the world," says Nicola Gampell, e-learning and marketing officer for Computer Aid International.

Based on the RACHEL-Pi (**magpi.cc/2fUKzua**), Connect will "bring them a local internet full of educational resources ranging from scientific simulations to Wikipedia articles," Nicola tells us.

With Computer Aid Connect kids will be able to access educational sites even without a connection

"All too often we're reminded of this reality," writes Jeremy Schwartz, executive director of World Possible, which created RACHEL. "There are places where young people aren't given the resources they need to learn. For many, the internet has become a small equalising force, but for many more, that equaliser does not exist.

"In 2017, we're going to test RACHEL against as many different use cases as we can," continues Jeremy. "We'll be formalising our own testing through our social entrepreneurs, and intimately supporting a narrower group of other organisations".

"We currently have 20 [devices] about to arrive to a project in Ethiopia and one in Mauritania," says Nicola. "So hopefully we'll be getting to see it in action soon.

"The Raspberry Pi is a key component of the device, being beneficial to for its low power usage and low cost.

"It also uses the UPS PIco Uninterruptible Power Supply to make it sustainable and stable during power outages."

Inside each Computer Aid Connect is a 64GB SD card and a wireless N150 high-power USB adapter (**magpi.cc/2fUW58N**).

"The version of the Raspberry Pi used changes between the Pi 2 and the old Model A," explains Nicola, "as we receive donations of old Raspberry Pi devices."

You can donate to Computer Aid, or request devices, on the charity's website (**magpi.cc/2fURnIo**).

SUSE GETS SERIOUS

A version of SUSE Linux Enterprise Server (SLES) has been released for Raspberry Pi.

SUSE is used by the European Space Agency to handle Mission Control, and by the Leibniz Supercomputing Centre to control SuperMUC, the fastest supercomputer in Europe.

In October, SUSE announced that it had optimised SLES for ARM-based 64-bit servers.

"SUSE Linux Enterprise Server for ARM will give customers more choice, flexibility, and opportunities," says Ralf Flaxa, SUSE president of engineering. "And they will be able to do it faster than ever before."

"We decided to bring SUSE Linux Enterprise Server to the Raspberry Pi to increase the visibility for SUSE and SLES," comments Jay Kruemcke, senior product manager at SUSE. "But to be honest, we also did it because it looked like fun.

"The real breakthrough for us in this process was the enthusiastic support that we received from Eben Upton when we told him of our plans," Jay tells us.

Right SUSE Linux Enterprise Server is used to run mission-critical software around the world

BRAND NEW EDUCATOR'S EDITION

The second edition of *The MagPi Educator's Edition* is now available. This special edition of *The MagPi* is created for teachers and educators interested in Raspberry Pi and computer science.

"At the heart of Raspberry Pi is a global community of educators who are working both inside and outside the classroom to inspire kids to get creative with technology," writes Philip Colligan, CEO, Raspberry Pi Foundation. "Our job is to provide that community with the support that they need.

"One of the ways we do that," explains Philip, "is by developing high-quality teaching resources and projects, many of which don't require a Raspberry Pi computer, all of which have been designed by educators, and all of which are available for free."

The MagPi Educator's Edition is freely licensed under Creative Commons and can be downloaded for free (**magpi.cc/2fVoPLB**).

Left *The MagPi Educator's Edition* is a special edition designed for teachers and educators

NOW TRENDING

The stories we shared that flew around the world

HELLO DEUTSCHLAND

magpi.cc/2eJDd6X

The latest edition of *The MagPi* translated into German – by our friends at CHIP – is now available. Our German readers are loving this over on Facebook. Raspberry Pi Trading CEO Eben Upton says, "Germany is our third largest market, so it seems natural that Germany should be the first country to get a localised version."

SPARKFUN PIRETROCADE

magpi.cc/2eJO9RZ

Our Google Plus crowd liked the look of this arcade joystick control kit.

PI ZERO PC STICK

magpi.cc/2eJMqMv

Over on Twitter, people got very excited at this project, by Nova Spirit, to turn a Pi Zero into a stick PC. Very few components are required for this 'PC in your pocket'.

UBUNTU CORE 16

Latest version of Ubuntu's Snappy software introduces a raft of new features

Ubuntu Core 16, also known as 'Snappy Ubuntu Core', has been released for the Raspberry Pi.

The latest version of Snappy Ubuntu Core has a completely upgraded software packaging system.

"In Ubuntu Core 16 we keep all of the software as compressed and signed files," Mark Shuttleworth, Canonical CEO told *The Inquirer* (**magpi.cc/2fUTMCK**).

Right Snappy Ubuntu Core 16 now on the Raspberry Pi

"Hackers can't modify that software on the disk, and the software on the device can always be validated," explained Mark.

On Ubuntu Core 15 the individual files were spread out over the disk, but in Core 16 they are concentrated into a single blob. Snap updates are transactional, which means that failures are automatically rolled back, giving developers the confidence to update their applications regularly.

The files exist as immutable squashFS blobs on disk. This means that a Raspberry Pi running Ubuntu Snappy Core can store different versions of each file used in the system. Files can be automatically updated, choosing the best fit for the system. Updates are automatically rolled back if things don't go to plan.

"The snap mechanism really changes the game," said Mark. "Your lawnmower will update itself every day, your MRI device in a hospital can be presented with updates offline, cars and robots can update themselves in a very reliable fashion, but they can also get new software installed on them that gives them new capabilities,"

You can download the Snappy Ubuntu Core 16 ISO on Ubuntu's website (**magpi.cc/2fV43Pn**).

ASTRO PI WINS ARTHUR CLARKE AWARD

Prestigious prize awarded to programme manager, Dave Honess

The Sir Arthur Clarke Award for Space Achievement has been awarded to Dave Honess, programme manager for Astro Pi. The British Interplanetary Society (BIS) presented the award at a glittering reception, held at the Reinventing Space Conference Gala Dinner at the Royal Society in London. Dave received the Award for Space Achievement – Industry/Project Individual.

Below Winners of the Sir Arthur Clarke Awards at the Royal Society in London

"It's a great honour to receive this award," said Dave. "I was rather surprised because the other nominees had all achieved so much... Astro Pi has had an impact on our whole organisation and because my name is on the trophy, I would like to pay tribute to all of my colleagues at Raspberry Pi who helped make it all happen. That's basically everyone! We couldn't have done it without the UK Space Agency, ESA, and a whole host of aerospace companies that gave us free lab time to qualify the payload for space."

"It's been great to work with Tim Peake, who's been hugely supportive from the start," Dave told us afterwards, "and while his flight is over, we're looking forward to the upcoming flight of French ESA astronaut Thomas Pesquet who will be expanding participation in Astro Pi to all ESA member states. My hope for the future is that Tim will fly again and, you never know, we might build him a new educational payload of some kind."

METERMAID MONITOR

Keep an eye out for traffic wardens with this intelligent detector

Like the rest of us, John Naulty, a dev ops engineer at Nuage Networks, hates getting parking tickets. But while most people just grumble, John actually built a machine-learning Raspberry Pi device that saves him from getting fined.

Known as Metermaid Monitor, the device is a speed camera that photographs passing cars and detects traffic wardens (known as 'meter maids' in the US).

"Have you recently moved into a neighbourhood and are afraid to bring your car for fear of the oppressive meter maid?" asks John. "I feel your pain."

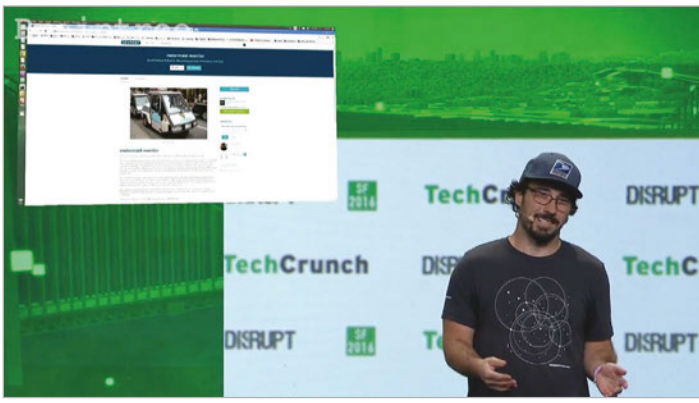
You can only park for two hours in San Francisco – a rule that's enforced by meter maids driving around in highly visible three-wheeled vehicles called 'Interceptors'. Meter maids note parked cars and come back every two hours. If the car is still around, they issue a fine.

Metermaid Monitor uses TensorFlow (**magpi.cc/2fZKtkl**), an open-source software library for machine intelligence, to spot the distinctive Interceptors. The TensorFlow instance was trained with images of Interceptors which were obtained using Google Image Search.

Car Speed Detector by Greg Barbu (**magpi.cc/2fZFMXA**) is used to detect and photograph vehicles driving past John's car. A Raspberry Pi with a Camera Module is used to run all the software.

"The purpose of this project was to provide a lazy way to avoid paying parking tickets," says John, "People who park in residential areas desire to park for as long as possible without having to move every two hours."

You can watch John Naulty explain his Metermaid Monitor at the Disrupt SF Hackathon 2016 (**magpi.cc/2fAQ9y6**).



Above John Naulty explained how the Metermaid Monitor system works at the Disrupt SF Hackathon

PROTECT YOUR EMPIRE AGAINST
CYBER ATTACKS
MANAGED CYBER PROTECTION
24 HOUR
Protect your Family, Finances & Files at Home or On the Go
DEVELOPED BY THE INDUSTRY EXPERTS
idappcom
LINUX OPERATING SYSTEM • INDUSTRY STANDARD INTRUSION DETECTION ENGINE • SMALL WEB SERVER • DNS SERVER • ENHANCED WIRELESS FUNCTIONS
HIPSY
HOME INTRUSION PREVENTION SYSTEM
GIPSY
GLOBAL INTRUSION PREVENTION SYSTEM
TRAVEL CYBER SHIELD
ANYTIME, ANYPLACE
MANAGED CYBER-PROTECTION SERVICE
SHIELD FROM EXPLOITS
Attackers working from outside your system
PROTECT FROM PHISHING
Invitations to reveal your private information
BLOCK MALWARE
Attackers that have been installed inside your system
USE THE FORCE OF IDAPPCOM
Our products are built on the best of British small computer, using our decades of expert security experience. We provide 24 hour security management via the cloud connected Idappcom Security Operations Centre.
Taking the Raspberry Pi to the Next Level of Cyber Protection
Visit: www.ipssecurityrules.com Call: +44 (0) 203 355 6804

A very Raspberry Pi Christmas

Power up the festive season with a Raspberry Pi and some holiday magic

We love Christmas here at *The MagPi*. The food, the atmosphere, the food, the decorations, and even the food. Putting up the decorations for Christmas is a time-honoured tradition, whether they be lights and baubles on a tree, snowman figures, or dancing rock 'n' roll reindeers that scare the dog.

Every year we say to ourselves that we want to make something a little special with our ready access to maker tools. Lights, code, and Raspberry Pis are all you need to create some cheap yet spectacular displays. You may have noticed that we've designed the cover this issue so that you can make it one of your displays, and we'll go through how to do that over the next few pages.

As well as creating a Christmas decoration with the magazine, we've got a couple of other projects you can do, including real Christmas lights and a fun Santa tracking system you can code yourself on Raspberry Pi.

16 **MAGAZINE TREE**
Turn this magazine into your newest Christmas decoration

24 **SANTA TRACKER PI**
Follow Saint Nick as he makes his way around the world

DIY CHRISTMAS TREE
Make your real Christmas tree better with some NeoPixels and code
22

CHRISTMAS CRACKERS
Want some Raspberry Pi-themed gift ideas? We have you covered...
26

SEND US YOUR PHOTOS!

We'd love to see what you've made with this issue's cover. You can email them to us at **magpi@raspberrypi.org** or tweet them to us **@TheMagP1**

The MagPi Christmas tree

TURN THE COVER INTO A CHRISTMAS DEC!

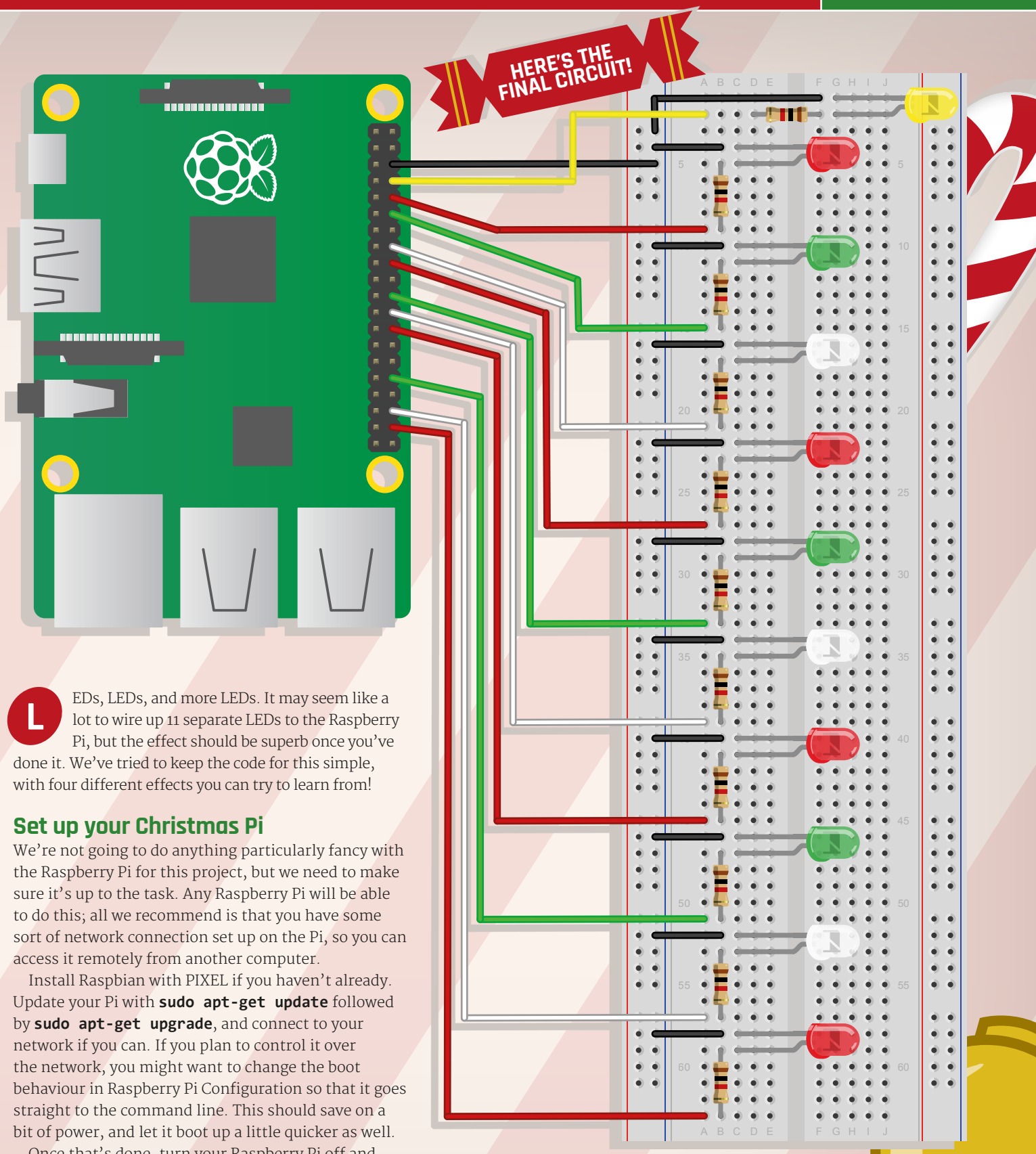
Make a light-up tree out of the cover for a unique decoration this year

ALTERNATE COVER

We know you all probably want to keep your copy of *The MagPi* 52 intact so you can reread the articles and tutorials at your leisure. While we wouldn't mind you carving up the cover, if you want to do so without permanently damaging your physical copy, remember that every issue of *The MagPi* is available as a free PDF on our website. Get some strong card, print off the cover from the PDF, and you'll be able to have the best of both worlds.

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
Issue 52 December 2016
raspberrypi.org/magpi
MERRY CHRISTMAS MAKES
Get creative with four festive projects
AMIGA POWER!
BEGINNER ELECTRONICS
BUILD A DIY ALARM CLOCK
GIVE YOUR ROBOT BRAINS
THE ONLY PI MAGAZINE WRITTEN BY THE RASPBERRY PI COMMUNITY

YOUR OFFICIAL RASPBERRY
The MagPi
The official Raspberry Pi magazine
Issue 52 December 2016
raspberrypi.org/magpi
MERRY CHRISTMAS MAKES
Get creative with four festive projects
AMIGA POWER!
Turn your Pi into the ultimate emulation station
BEGINNER ELECTRONICS
Get started with prototyping on your Pi
BUILD A DIY ALARM CLOCK
Wake at dawn with a sunrise alarm
GIVE YOUR ROBOT BRAINS
How to add a wealth of sensors to your Pi robot
Issue 52 • Dec 2016 • £5.99
THE ONLY PI MAGAZINE WRITTEN BY THE RASPBERRY PI COMMUNITY



LEDs, LEDs, and more LEDs. It may seem like a lot to wire up 11 separate LEDs to the Raspberry Pi, but the effect should be superb once you've done it. We've tried to keep the code for this simple, with four different effects you can try to learn from!

Set up your Christmas Pi

We're not going to do anything particularly fancy with the Raspberry Pi for this project, but we need to make sure it's up to the task. Any Raspberry Pi will be able to do this; all we recommend is that you have some sort of network connection set up on the Pi, so you can access it remotely from another computer.

Install Raspbian with PIXEL if you haven't already. Update your Pi with `sudo apt-get update` followed by `sudo apt-get upgrade`, and connect to your network if you can. If you plan to control it over the network, you might want to change the boot behaviour in Raspberry Pi Configuration so that it goes straight to the command line. This should save on a bit of power, and let it boot up a little quicker as well.

Once that's done, turn your Raspberry Pi off and wire up the circuit as shown in the diagram to test it. There are ten LEDs for the lights on the tree, and an eleventh LED that we're using for the star. Follow the next page on how to wire up the lights to the cover...

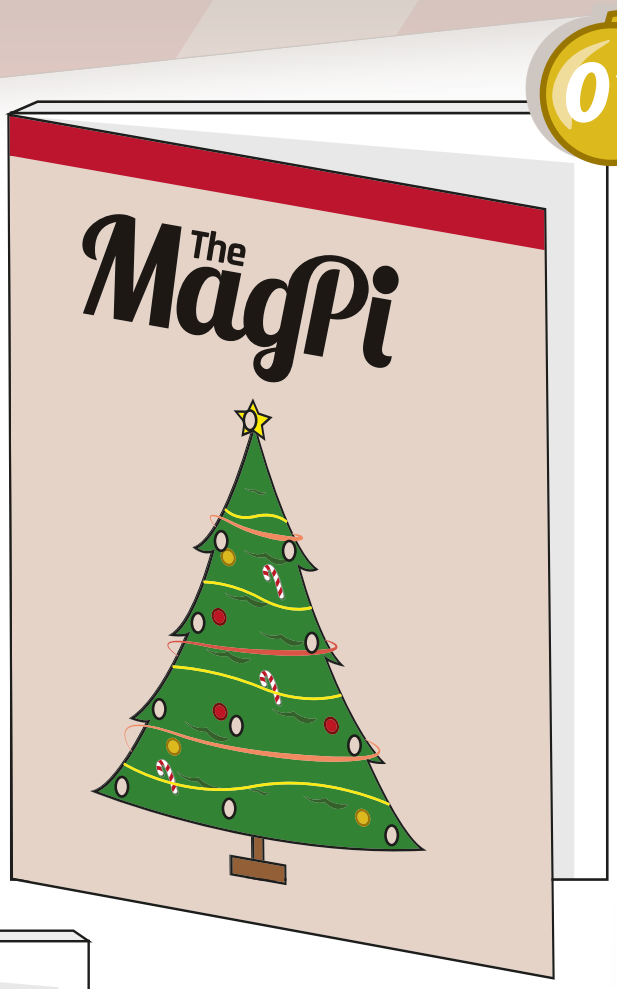
You can use a breadboard to make wiring easier or you can wire up the LEDs directly to the Pi. Use any colours you wish!

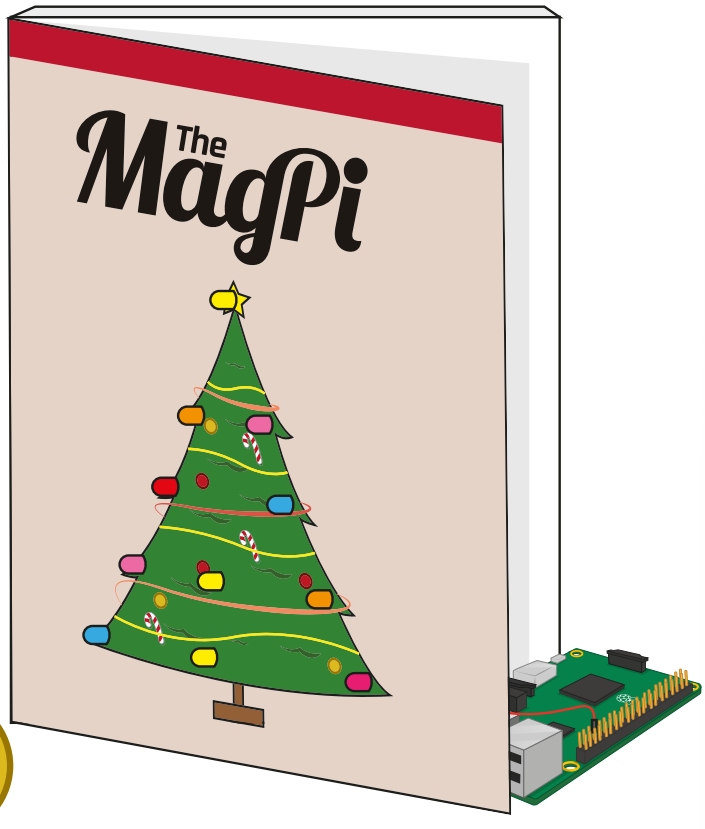
You'll Need

- Raspberry Pi (any model)
- 11× 5mm LEDs (any colour)
- 11× 1kΩ resistors
- Jumper wires
- Wireless network connection (optional)

>STEP-01
Cut out the holes

We don't usually condone defacing the cover of *The MagPi* – we do work hard on making it look fantastic for you! In this case, though, we've designed it so you can cut out little LED holes on the cover. Open up the magazine so the cover is laying flat against a cutting board. Use a craft knife or box cutter and carefully cut holes for the LEDs on the cover. If you're feeling especially handy, you can always use a 4mm drill. Make the holes slightly smaller than the LEDs.





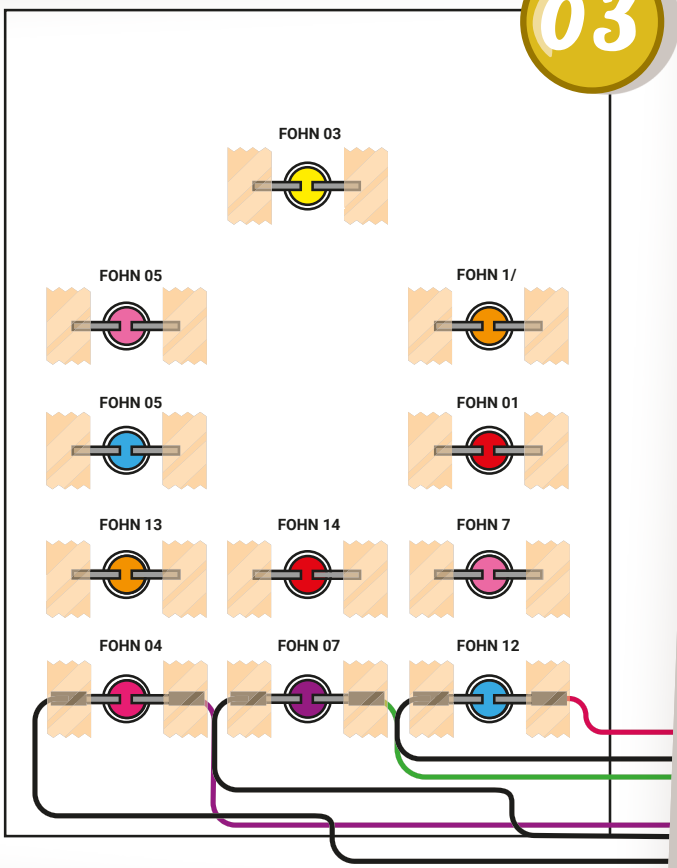
>STEP-02
Push through the LEDs

If your holes aren't the perfect size or shape, don't worry, as you'll want them to hold the LEDs in the cover. Push them through the holes from the back; with the holes a little smaller, they should be snug. You can arrange them in any order you want, with any colour you desire; however, you may need to change some of the code later to accommodate this (and we'll show you how).

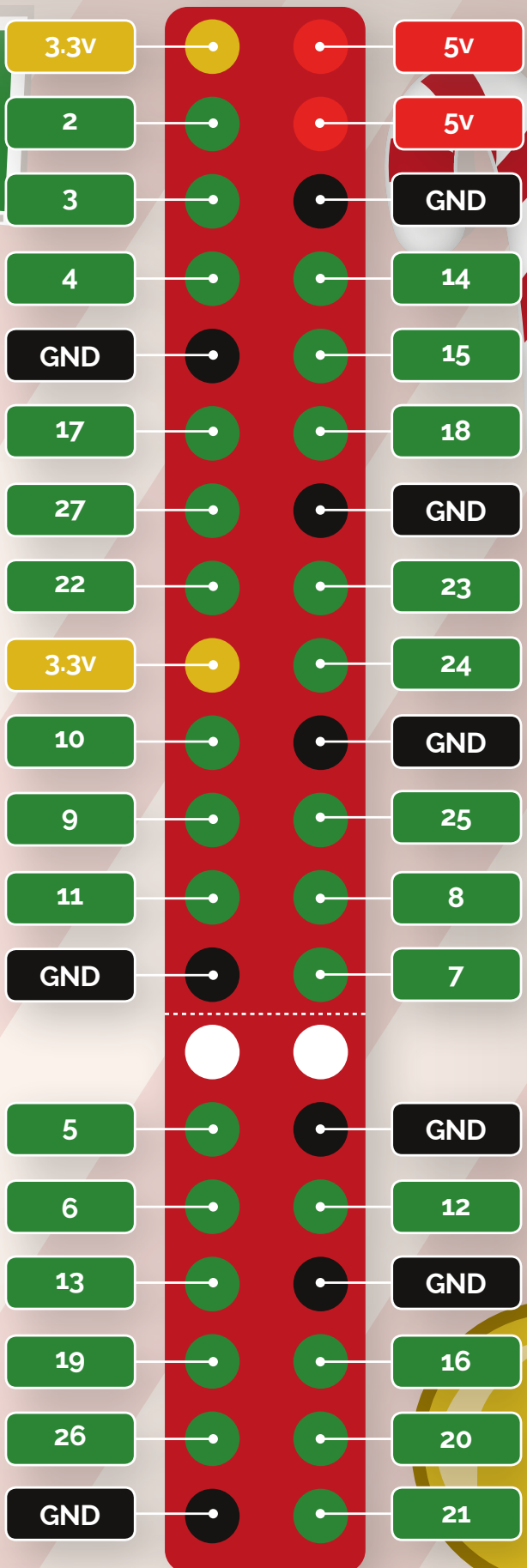
>STEP-03

Affix the lights

We recommend you bend back the LED pins so there's a little bit more room for the cover to close. Make sure the shorter pins (the ground pins) are all bent in the same direction to make the wiring easier. Add little strips of sticky tape to each leg to keep them in position, and then you can start attaching jumper wires. Refer to the diagram on the previous page when wiring them back up to the breadboard.



Just in case you need to change the code, here's a reference for the GPIO pins



>STEP-04

Display it

What good is a decoration if it doesn't have a place to live? You'll have to consider a place to stand the magazine, a way to stand it up (clear acrylic magazine stands are pretty cheap on eBay), and distance to a power source. Get it all programmed up over the page, get it in position, and your newest and (in our opinion) greatest decoration will be ready.

FIND THE CODE OVER THE PAGE!

Download all the code here: magpi.cc/XmasCover

Lights on

This is the most basic script for the cover that you can use. We've decided to use GPIO Zero to make coding this a little easier. The **star** variable is what we're using for the star on top of the tree, which in our circuit is attached to GPIO 14 (pin 8). The rest of the LEDs are attached to the programmable pins on the edge of the Pi, which represent the GPIO pins listed in the variable **lights**. Refer to the GPIO pins on the previous page if you need to identify them.

We then turn the star LED on, and then turn on all the other lights, using a **for** loop to activate them all in sequence with minimal code.

lightson.py

```
#!/usr/bin/env python

from gpiozero import LED

star = LED(14)

lights = [LED(15), LED(18), LED(23), LED(24),
          LED(25), LED(8), LED(7), LED(12),
          LED(16), LED(20)]

star.on()

for i in range(10):
  lights[i].on()
```

Twinkling lights

With this code, we've modified it so that the LEDs that make up the tree lights will randomly turn on and off as they go, leaving seven on at all times. As will happen with the rest of the code, the star LED code will remain unchanged.

We turn on seven lights in a similar fashion, then create a **while** loop to turn off a random light that's on, and replace it with a light that's off.

twinkling.py

```
#!/usr/bin/env python

from gpiozero import LED
from random import randint
from time import sleep

star = LED(14)

lights = [LED(15), LED(18), LED(23), LED(24),
          LED(25), LED(8), LED(7), LED(12),
          LED(16), LED(20)]

star.on()

for i in range(7):
  lights[i].on()

while True:
  turn_off = randint(0,9)
  turn_on = randint(0,9)
  if lights[turn_off].is_lit == True and
lights[turn_on].is_lit == False:
    lights[turn_off].off()
    lights[turn_on].on()
    sleep(0.5)
```

candle_flicker.py

```
#!/usr/bin/env python

from gpiozero import PWMLED, LED
from gpiozero.tools import random_values
from signal import pause

star = LED(14)

lights = [PWMLED(15), PWMLED(18), PWMLED(23),
          PWMLED(24), PWMLED(25), PWMLED(8),
          PWMLED(7), PWMLED(12), PWMLED(16),
          PWMLED(20)]

star.on()

for i in range(10):
  lights[i].source = random_values()

pause()
```

Wave lights

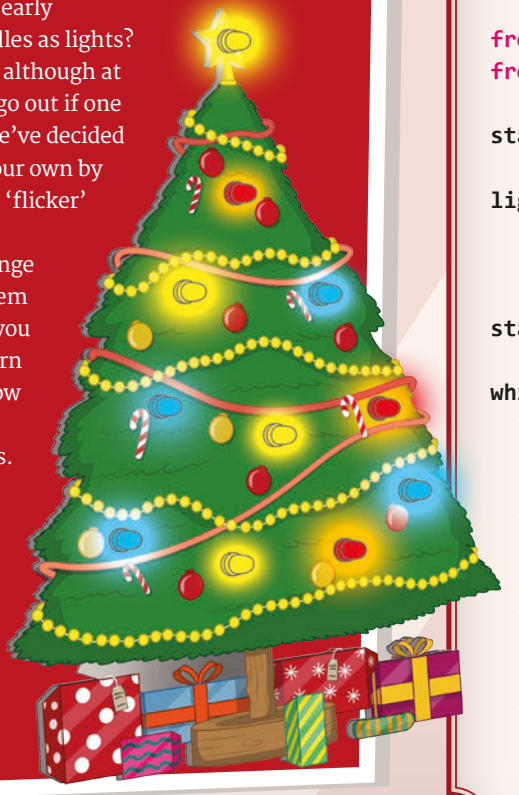
Create a Mexican wave on the cover! This is much harder to do on a real tree, so we thought we'd give it a go here first. It works very similarly to our previous code examples, but this time it only activates so many lights at a time. We use the **toggle** function of GPIO Zero to toggle LEDs on or off, which also allows us to make the code more efficient.

This may involve a little trial and error, though, to make sure it's all working well with your light arrangement.

Candle lights

Did you know that a lot of early Christmas trees used candles as lights? Bit of a fire hazard, really, although at least the whole lot didn't go out if one of the bulbs went dead. We've decided to create candle lights of our own by programming the LEDs to 'flicker' like candles.

First, you'll need to change the **lights** list to have them all listed as PWMLEDs. If you followed our Jack-o-lantern tutorial last issue, you know we can use this to change the brightness of the LEDs. Using a slightly different command to turn them on, **lights[i].source = random_values()**, we can make them have a random PWM value. This will get you a flicker effect with a lot less code than last issue.



waves.py

```
#!/usr/bin/env python

from gpiozero import LED
from time import sleep

star = LED(14)

lights = [LED(15), LED(18), LED(23), LED(24),
          LED(25), LED(8), LED(7), LED(12),
          LED(16), LED(20)]

star.on()

while True:
  for i in range(3):
    lights[i].toggle()
  sleep(0.5)
  for i in range(3):
    lights[i+3].toggle()
  sleep(0.5)
  for i in range(2):
    lights[i+6].toggle()
  sleep(0.5)
  for i in range(2):
    lights[i+8].toggle()
  sleep(0.5)
```

START AT BOOT

You could use SSH to connect to your Pi each time and run the scripts, or you can have them run at startup. To do that, open up the profile file with **sudo nano /etc/profile** and add a line like this to the bottom:

```
sudo python lightson.py
```

Neopixel Tree lights

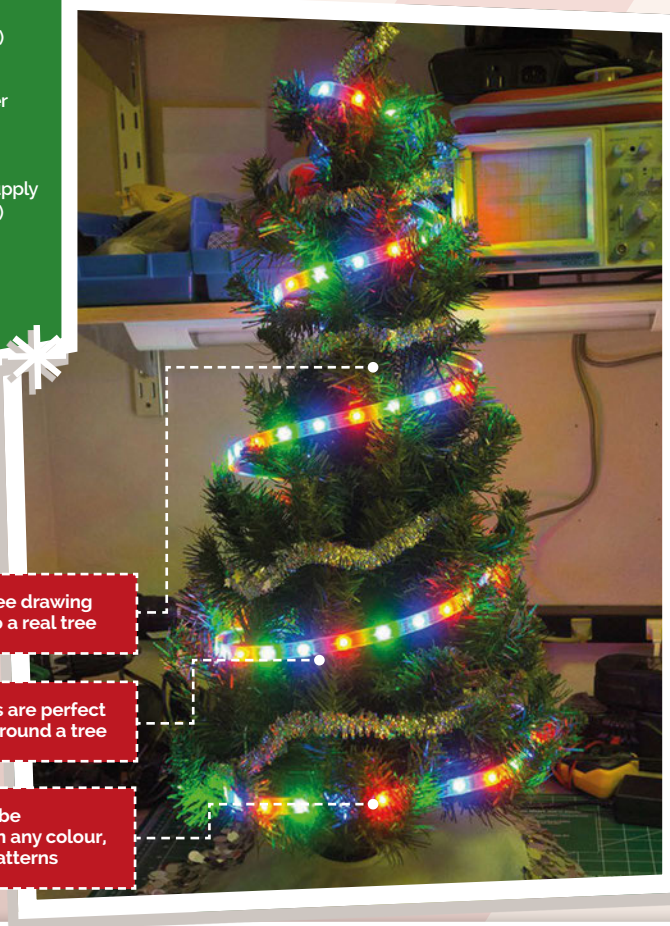
Make your own turbo-charged fairy lights for a real Christmas tree

You'll Need

- Pi Zero
- RGB LED NeoPixel strip – enough length to wrap around your tree
- Level converter chip 3v - 5v (74AHCT125)
- Female DC plug adapter **magpi.cc/2eD5eR3**
- 5V power supply (for DC plug)
- Push button
- Wires, breadboard

Over the last few pages we've programmed and lit up the cover. We hope you've enjoyed doing that (send us a tweet with a picture to **@TheMagPi**!), but it may have left you in the mood for a slightly bigger project.

This is where a strip of NeoPixels comes in. These are programmable RGB LED strips that easily connect up to a Raspberry Pi and can create some incredible displays. Let's make them Christmassy...



Go from the tree drawing on the cover to a real tree

NeoPixel strips are perfect for wrapping around a tree

The lights can be programmed in any colour, and to make patterns

>STEP-01

Get a Pi ready

NeoPixels work a bit better with first-generation Raspberry Pis, so we suggest using a Pi Zero or A+ to save on space and power when making your tree lights. Install the latest version of Raspbian with PIXEL, and perform the customary **sudo apt-get update** and **sudo apt-get upgrade** on it.

```
sudo apt-get install build-essential
python-dev git scons swig
```

Now download the library for the NeoPixels:

```
git clone https://github.com/jgarff/rpi_
ws281x.git
cd rpi_ws281x
scons
```

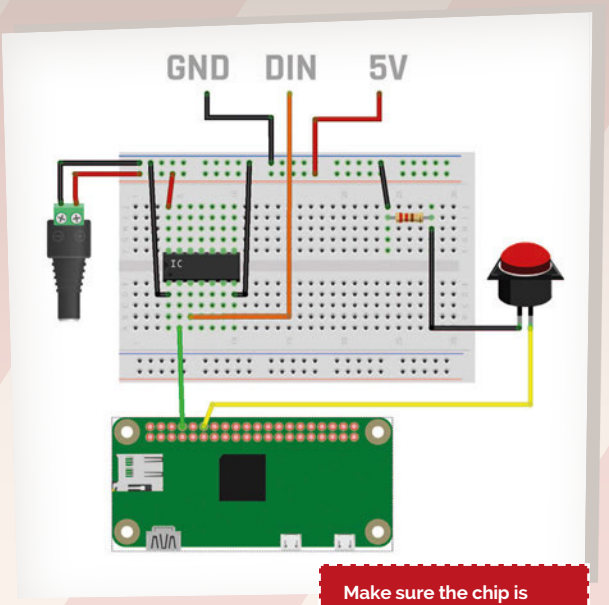
Finally, we can install the module on the Pi:

```
cd python
sudo python setup.py install
```

>STEP-02

Test circuit

It will be easiest to build this up on a breadboard first. Follow the Fritzing diagram to do this. The strip is powered by an independent power source (the Pi can't provide enough power to light the LEDs) and the signal from the Raspberry Pi is converted through the chip so it can be properly read by the strip. The NeoPixel strip sometimes comes with a multi-prong connector; you only need to use one of the wires to program the LEDs. If you look carefully, this wire will be connected to the middle DIN connector on the strip, and will very likely not be a red or black wire.



Make sure the chip is wired up properly so the signals can be converted

>STEP-03
Get the code

Type up the code listing for this tutorial or download it to your Raspberry Pi. It's Python code using a library that ports the normally Arduino code over to Python and the Pi. With the circuit wired up and the Pi switched on, run the code and see if it's all working properly.

When initially switched on, the first 30 LEDs should turn white. Pressing the push button will create a series of red and green lights that run down the strip in an endless loop. Finally, the last press will make them slowly pulsate with rainbow-coloured lights. Pressing the button again will renew the loop.

>STEP-04
Deck the tree

Once you've done all your testing, it's time to decorate the tree. It might be best to disconnect the NeoPixels from the circuit before continuing. Wrap the tree with the NeoPixels (with the end at the bottom), and then cut between LEDs where you want the strip to end. If you have a waterproof covering, you can remove it now.

Plug it all back in and do another test. You'll now want to change the **LED_COUNT** number to represent the actual number of NeoPixels in the strip around your tree; you can either count them or use trial and error!

>STEP-05
Build a better circuit

It might be a good idea to construct the circuit properly if you're worried about the breadboard falling apart. The chip can be substituted for a diode which may make it easier (check the Adafruit website: **magpi.cc/2fvshuy**). You probably don't want the lights on all the time either; if you want the program to run at boot, check out the boxout in the previous tutorial on how to get Python scripts to run at boot. Most importantly, though, enjoy your new lights!

Neopixel_tree.py

Language >PYTHON

DOWNLOAD: magpi.cc/NeoPixelTree

```
from neopixel import *
from gpiozero import Button
from time import sleep

button = Button (18)

LED_COUNT      = 30      # Number of LED pixels
LED_PIN        = 14      # GPIO pin
LED_FREQ_HZ    = 800000  # LED signal frequency in hertz
LED_DMA        = 5
LED_BRIGHTNESS = 255     # Brightness of LEDs
LED_INVERT     = False

def lights_on (strip, color):
  for i in range(strip.numPixels()):
    strip.setPixelColor(i, color)
    strip.show()

def lights_wave(strip, color1, color2):
  for q in range(3):
    for i in range(0, strip.numPixels(), 3):
      strip.setPixelColor(i+q, color1)
    strip.show()
    sleep(0.2)
    for i in range(0, strip.numPixels(), 3):
      strip.setPixelColor(i+q, color2)

def color_wheel(pos):
  if pos < 85:
    return Color(pos * 3, 255 - pos * 3, 0)
  elif pos < 170:
    pos -= 85
    return Color(255 - pos * 3, 0, pos * 3)
  else:
    pos -= 170
    return Color(0, pos * 3, 255 - pos * 3)

def lights_rainbow(strip):
  for j in range(256):
    for i in range(strip.numPixels()):
      strip.setPixelColor(i, color_wheel(
((i * 256 / strip.numPixels()) + j) & 255))
      strip.show()
      sleep(0.1)

strip = Adafruit_NeoPixel(LED_COUNT, LED_PIN, LED_FREQ_HZ,
                          LED_DMA, LED_INVERT, LED_BRIGHTNESS)
strip.begin()

while True:
  lights_on(strip, Color(255,255,255))
  button.wait_for_press()
  sleep(0.5)
  while button.is_pressed == False:
    lights_wave(strip, Color(255,0,0), Color(0,0,255))
  sleep(0.5)
  while button.is_pressed == False:
    ligthts_rainbow(strip)
  sleep(0.5)
```

Feature

Santa Tracker Pi

Find out where Santa is on his journey around the world by visualising it on a map

You'll Need

- Pygame Zero
- Internet access on your Pi
- Art assets **magpi.cc/SantaTrackerPi**

Father Christmas has got to make a lot of stops on Christmas Eve, but he somehow manages to do it anyway. He's an impressive fellow; very spry for his age. What you might not know, though, is that it's possible to track him throughout the night. NORAD uses its global satellite and radar systems to track Santa, but we have to make do with a little lower-tech solution on the Pi. Let's get to work.

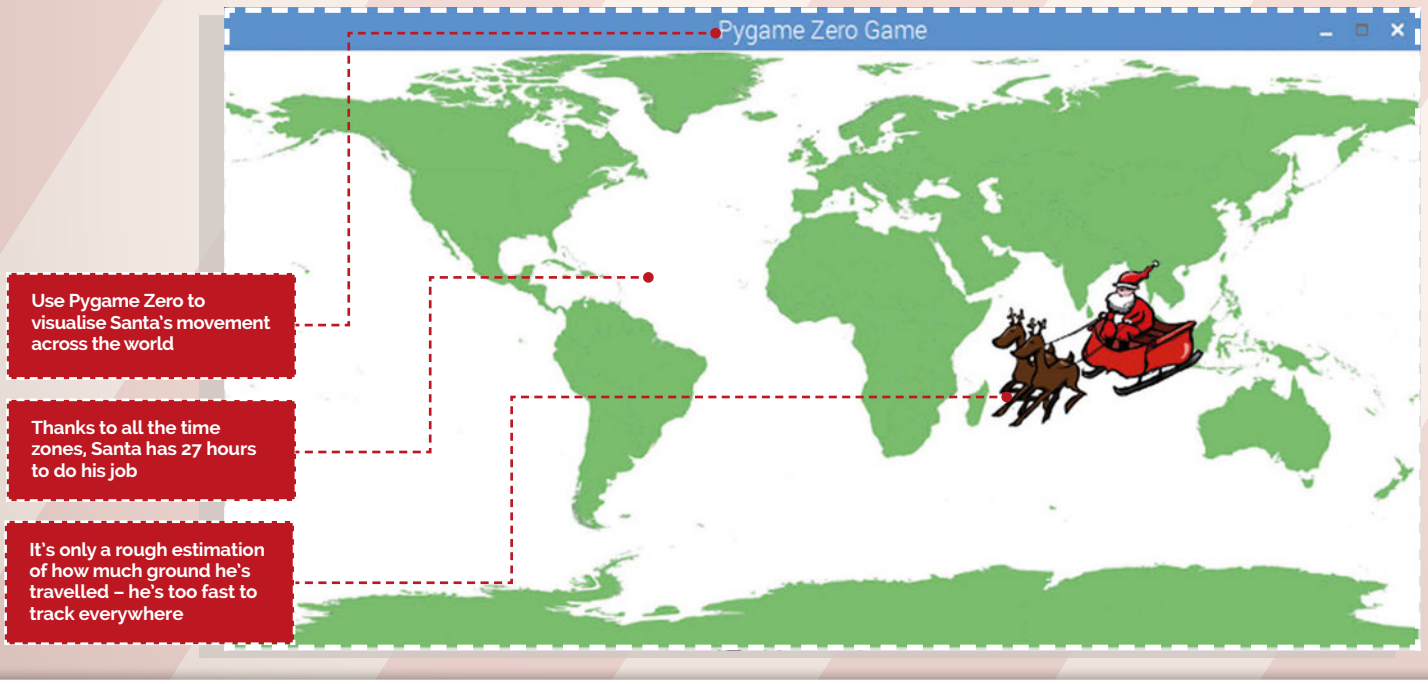
For this project we're going to use Pygame Zero; it's a very easy way to create graphical representations on the Raspberry Pi. It's already installed on Raspbian with PIXEL and makes use of the Pygame library to work. Make sure your Raspberry Pi is up to date with a quick **`sudo apt-get update`** followed by **`sudo apt-get upgrade`** in the terminal.

You'll also want to make sure your Raspberry Pi boots to the PIXEL desktop, as we'll be doing this all in a graphical environment. If you haven't already, connect your Raspberry Pi to the internet, either via wireless or a wired Ethernet cable. For this code we require a constant connection to the internet.

You'll also need to get the art assets for the program. Download them from **magpi.cc/SantaTrackerPi** and put them in a folder called **images** – we just need the two.

Tracking Saint Nick

Although we don't have access to the same surveillance technology as the folks at NORAD, we can still make a pretty good approximation as to where Santa Claus will be at any given time. Since he always arrives at midnight, we can make an educated guess that wherever it's midnight in the world is where he'll be.


Pygame Zero Game
Use Pygame Zero to visualise Santa's movement across the world
Thanks to all the time zones, Santa has 27 hours to do his job
It's only a rough estimation of how much ground he's travelled – he's too fast to track everywhere

Different parts of the world operate at different times. San Francisco is eight hours behind the UK, and Tokyo is nine hours ahead. When it's midnight in Japan, it's only 3pm in the UK. With that in mind, we can take into account the full range of time zones, and calculate how far around the planet Santa has been at any point. Santa starts as early as 10am UK time on Christmas Eve, in the furthest reaches of the East, and finishes at midday Christmas Day at the International Date Line.

Clever code

We now know enough to make our program. Type it up or download it and save it as **tracker.py**; this and the **images** folder should both be in the same folder together. Here's how it works.

First, we set the stage. We have our Santa sprite, which we use the **Actor** class on so the code knows it's a character sprite. We also set the width and height of the window. Once that's done, we draw both the map image background and Santa onto the screen.

Santa's position is worked out by first finding out the time. We want to use Coordinated Universal Time (UTC), which is a constant. Luckily, it's the same time as GMT, so if you live in the UK it's a bit easier to understand. By finding it online using **urllib**, we know the exact time and don't have to worry about any inaccuracies in the Raspberry Pi's clock. We then parse it to remove the HTML from it, and look for the month, day, hour, and minute. This code will only work on Christmas Eve and Christmas Day, so we need to know exactly when the program is running.

We then use **if** statements which check to see if it's the right time of year: after 10am on Christmas Eve UTC, and before midday Christmas Day UTC. If it's not, we assume he's about to start or has just finished, so we make his position reflect that using the pixel coordinates of the window we made. 0 is the far left, and 800 is the far right.

If it is a time during his journey, we then calculate roughly where he is. Santa will be working over 27 hours, so we know that each hour after 10am he'll have covered 1/27th of the map. We then also add a little bit, depending on the minute; this means the map will be more accurate, and also it'll update every minute so you can track his progress.

Run the script

That's all we need to do. Pygame Zero automatically loops the script a few times a second, so it'll update live as Santa makes his way across the world. To run it, open a terminal, use **cd** to move to the folder with **tracker.py** and the **images** folder, and then use:

```
pgzrun tracker.py
```

Remember to be tucked up in bed when he comes to visit you, though, or you might not get any presents!

tracker.py

Language
>PYTHON

DOWNLOAD:
magpi.cc/SantaTrackerPi

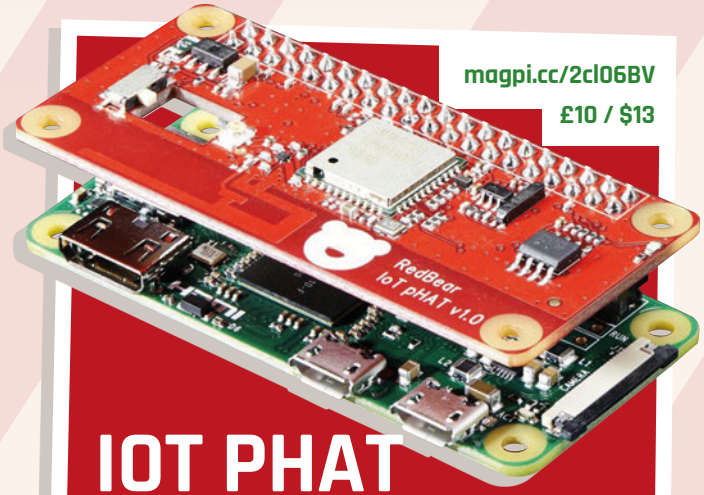
```
import urllib.request

santa = Actor('santa')

WIDTH = 800
HEIGHT = 400

def draw():
  screen.blit('map',(0,0))
  santa.draw()

def santa_loc(c_day, c_hour, c_minute):
  if c_day == 24:
    location1 = (WIDTH/27)*(c_hour - 10)
    location2 = ((WIDTH/27)/60)*(c_minute)
  else:
     location1 = (WIDTH/27)*(c_hour + 14)
     location2 = ((WIDTH/27)/60)*(c_minute)
  santa.pos = (WIDTH - (location1 + location2)), (HEIGHT/2)


def update():
  response = urllib.request.urlopen (
'http://just-the-time.appspot.com/')
  web_time = (response.read()).decode('utf-8')
  c_day = web_time[0:2]
  c_month = web_time[3:5]
  c_hour = int(web_time[11:13])
  c_minute = int(web_time[14:16])

  if c_month == 12 and c_day == 24:
    if c_hour >= 10:
      santa_loc(c_day, c_hour, c_minute)
    else:
      santa.pos = (WIDTH), (HEIGHT/2)
  elif c_month == 12 and c_day == 25:
    if c_hour <= 12:
      santa_loc(c_day, c_hour, c_minute)
    else:
      santa.pos = (WIDTH-WIDTH), (HEIGHT/2)
  else:
    santa.pos = (WIDTH), (HEIGHT/2)
```

Christmas Crackers

Amazing ideas for Raspberry Pi gifts this Christmas

Christmas is the time for giving (even if you're giving to yourself). And what more could a self-respecting Raspberry Pi maker want than one of the gifts here?

We've scoured the Raspberry Pi market for some of the best gifts for a Raspberry Pi lover. From environment-sensing pHATs to portable games consoles, there's something here for everyone. We've kept the prices low, so buying a Raspberry Pi as a gift won't break the bank. And remember you can pick up a Pi Zero for just £4 / $6. You can check who has Pi Zero boards at **magpi.cc/2fMveZh**.



magpi.cc/2cl06BV

£10 / $13

IOT PHAT

Add WiFi and Bluetooth to a Pi Zero

This small pHAT makes it easy to add WiFi and Bluetooth to a Pi Zero. In fact, it uses the same wireless radio chip as the Raspberry Pi 3. It's an excellent way to add some of our favourite Pi 3 functionality to the Pi Zero.

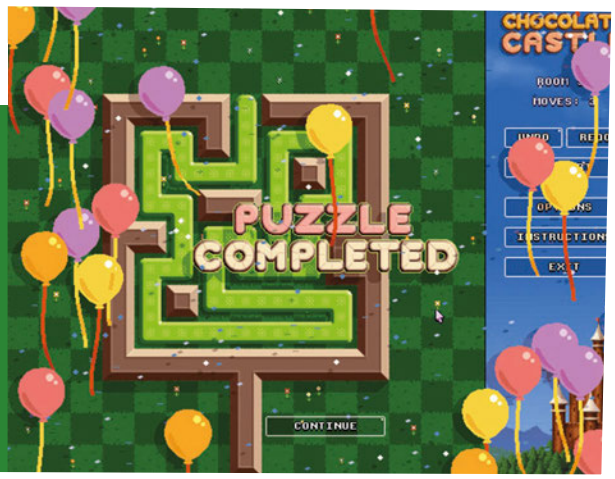
PICO-8

Fantasy Console Software

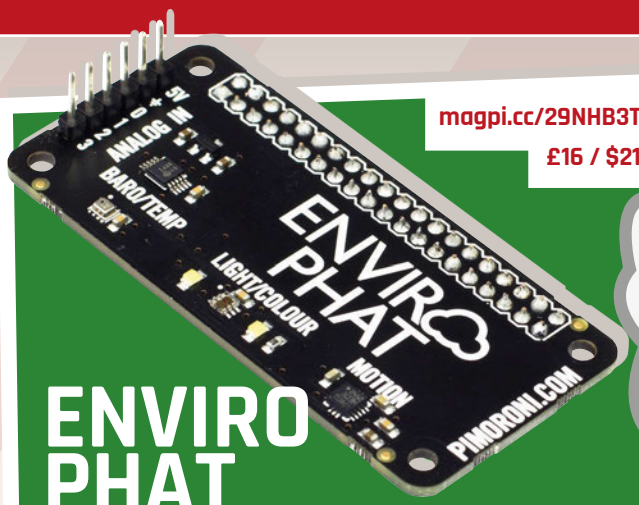
magpi.cc/2fMhLkn

£12 / $18

PICO-8 is software that turns your Raspberry Pi into an amazing retro games console. It's rare to pay for software on the Raspberry Pi, but we think PICO-8 is worth it. There is a huge community providing games, and you can stop any game (at any time) and change the graphics, sound effects, and music using built-in tools. You can then learn to make your games. Great fun!



Stocking Fillers



ENVIRO PHAT

magpi.cc/29NHB3T
£16 / $21

Take environment and motion measurements

The Enviro pHAT is packed with sensors for your Pi Zero (or any other Pi model). These enable you to measure temperature, pressure, light level, colour, motion, compass heading, and analogue inputs. It's like a smaller Sense HAT for your Pi Zero.

MUG

magpi.cc/2fIX8wA
£8 / $10

Circular receptacle with convenient handle for liquid transportation

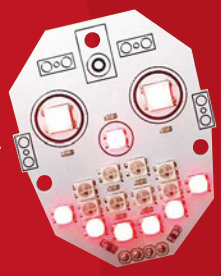
Tea (or coffee) and project-making go hand in hand. This mug can hold 330ml of your chosen beverage. It's bright red with a Raspberry Pi logo on the side and it's available along with other Pi goodies at the official Raspberry Pi Swag Store. Just be careful not to spill liquid on your Pi.

Great gifts for under £10

MCROBOFACE

A fun LED face that lights up to create different expressions. Make your robots happy, sad, or shocked. Great fun for Christmas Day.

£7 / $9
magpi.cc/2dXcipD



ZEROVIEW

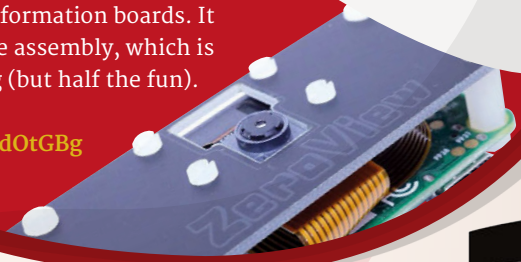
This clever camera mount screws onto your Pi Zero and has suction cups to stick it to a window. Ideal for shooting time-lapse videos.

£7 / $9
magpi.cc/2e89hWt

ZEROSEG

This 8-character display for your Pi can be used to create news tickers, time and date displays, and mini information boards. It does require assembly, which is challenging (but half the fun).

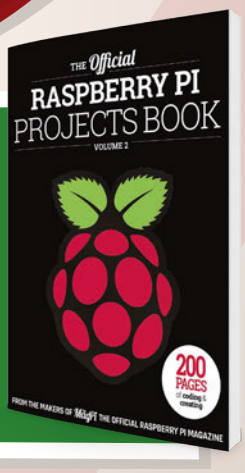
£10 / $13
magpi.cc/2dOtGBg



PROJECTS BOOK 2

Fresh from *The MagPi*, the *Official Raspberry Pi Projects Book* is 200 pages stuffed full of Raspberry Pi goodness, tutorials, projects, and features. Available from Amazon, Barnes & Noble, WHSmith, The Pi Hut, and the Swag Store.

£12.99
magpi.cc/Back-issues



FREE PI ZERO!

Subscribe in print for six or 12 months to receive this stunning free gift

Subscribe by 7th DEC 2016

to get your **FREE** PI ZERO in time for **Christmas!**

Subscribe today & receive:

- A free Pi Zero v1.3 (the latest model)
- A free Camera Module connector
- A free USB & HDMI cable bundle

Delivered with your first issue!

Other benefits:

- Save up to 25% on the price
- Free delivery to your door
- Exclusive Pi offers & discounts
- Get every issue first (before stores)

YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
The official Raspberry Pi magazine
MERRY CHRISTMAS MAKES
Get creative with four festive projects
BEGINNER ELECTRONICS
BUILD A DIY ALARM CLOCK
GIVE YOUR ROBOT BRAINS
THE ONLY PI MAGAZINE WRITTEN BY THE RASPBERRY PI COMMU
YOUR OFFICIAL RASPBERRY PI MAGAZINE
The MagPi
BUILD A REMOTE CONTROL ROBOT
Create your own Raspberry Pi-powered rover with our latest guide
MAKE AN UNDERWATER CAMERA
TOTALLY WIZARD
SEND YOUR CODE TO SPACE
Also inside:
SPOOKY HALLOWEEN PROJECTS
THE ONLY PI MAGAZINE WRITTEN BY THE RASPBERRY PI COMMUNITY
The MagPi
50
GREATEST RASPBERRY PI PROJECTS

Pricing

Get six issues:

£30 (UK)
£45 (EU)
$69 (USA)
£50 (Rest of World)

Subscribe for a year:

£55 (UK)
£80 (EU)
$129 (USA)
£90 (Rest of World)

Get three issues:

£12.99 (UK) (Direct Debit)
$37.50 (US) (quarterly)

How to subscribe:

- **magpi.cc/Subs1** (UK / ROW)
- **imsnews.com/magpi** (USA)
- Call +44(0)1202 586848 (UK/ROW)
- Call 800 428 3003 (USA)

Available on the App Store

GET IT ON Google Play

SUBSCRIPTION FORM

YES! I'd like to subscribe to The MagPi magazine and save money

This subscription is: ☐ For me ☐ A gift for someone*

Mag#52

YOUR DETAILS Mr ☐ Mrs ☐ Miss ☐ Ms ☐

First name ..... Surname .....

Address .....

.....

Postcode ..... Email .....

Daytime phone ..... Mobile .....

**If giving* The MagPi *as a gift, please complete both your own details (above) and the recipient's (below).*

GIFT RECIPIENT'S DETAILS ONLY Mr ☐ Mrs ☐ Miss ☐ Ms ☐

First name ..... Surname .....

Address .....

Postcode ..... Email .....

PAYMENT OPTIONS

1 DIRECT DEBIT PAYMENT £12.99 every 3 issues (UK only)
Instruction to your bank or building society to pay by Direct Debit

DIRECT Debit

Please fill in the form and send to:
The MagPi, Select Publisher Services Ltd,
PO Box 6337, Bournemouth BH1 9EH

Service user number 8 3 8 7 7 3

Name and full postal address of your bank or building society:

To: The Manager Bank/building society .....

Address .....

.....

..... Postcode .....

Name(s) of account holder(s) .....

Branch sort code ☐☐☐☐☐☐ Account number ☐☐☐☐☐☐☐☐

Reference ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ (Official use only)

Instruction to your bank or building society
Please pay Select Publisher Services Ltd Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit Guarantee. I understand that this instruction may remain with Select Publisher Services Ltd and, if so, details will be passed electronically to my bank/building society.

Signature ..... Date ☐☐/☐☐/☐☐

Banks and building societies may not accept Direct Debit instructions for some types of account.

SUBSCRIPTION PRICING WHEN PAYING BY CHEQUE OR CREDIT/DEBIT CARD

6 ISSUES ☐ UK £30 ☐ Europe £45 ☐ Rest of World £50

12 ISSUES ☐ UK £55 ☐ Europe £80 ☐ Rest of World £90

2 CHEQUE

I enclose a cheque for ..... (made payable to Select Publisher Services Ltd)

3 CREDIT/DEBIT CARD ☐ Visa ☐ MasterCard ☐ Maestro ☐ Switch

Card number ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiry date ☐☐ ☐☐ Valid from ☐☐ ☐☐ (if shown)

Issue number ☐☐ (if shown) Security number ☐☐☐
(last 3 digits on the back of the card)

Signature ..... Date ☐☐/☐☐/☐☐

I would like my subscription to begin from issue ..... (month + year)

RETURN THIS FORM TO:
MagPi Magazine Subscriptions, Select Publisher Services Ltd, PO Box 6337, Bournemouth BH1 9EH

☐ Please tick this box if you DO NOT want to receive any other information from Select Publisher Services Ltd.

☐ Please tick this box if you DO NOT want to receive any other information from other companies.

☐ Please tick this box if you DO NOT want to subscribe to The MagPi newsletter.

BRUCE SHAPIRO

Bruce grew up enchanted by music, electronics, and making things. He has devoted himself to using motion control for making art and education tools.
magpi.cc/2fUUaRN

Inside the table is a Raspberry Pi used to control the pattern being created

Beneath the sand surface is a robot called Sisbot. This moves the ball using a powerful magnet

A metal ball is pulled across a sand surface to generate a work of art

Quick Facts

- Each table is made in the US
- Bruce has been building Sisyphus tables for 20 years
- Sisyphus tables are displayed in museums and art galleries
- Components were picked to run quietly; it's virtually silent
- Sisyphus is demoed at Northeast Minneapolis makerspace **magpi.cc/2eWKiEH**

SISYPHUS

The incredible table that's also a work of art (and a Raspberry Pi robot)

Bruce Shapiro is a maker and an artist. Unlike Picasso or Rembrandt, Bruce doesn't paint with oil and brushes. "My medium is motion control," he tells us.

We caught up with Bruce to chat about his latest project, Sisyphus. Hot on the heels of a phenomenally successful Kickstarter campaign, Bruce has a lot to talk about.

"Sisyphus is a computer-controlled machine that moves a magnet beneath a field of sand," Bruce explains. "On the sand, a steel ball follows the magnet's changing position, creating dune patterns in its wake."

In Greek mythology, Sisyphus was condemned to roll a boulder up a mountain for all eternity. "In my art, Sisyphus is a kinetic sculpture that rolls a ball through sand, forever creating and erasing beautiful patterns." Bruce says that watching Sisyphus evokes a meditative feeling.

"Initially, I viewed Sisyphus as a kinetic sculpture. I still do! But over the years, I began to see a parallel with the relationship between making musical instruments and writing songs. As different as these creative skills are, both are absolutely integral to the final art."

Bruce has been creating Sisyphus sculptures for nearly 20 years, and has permanent installations in Switzerland, Germany, and Australia.

The heart of the project is the Sisbot, a robot that controls the metal balls which create the artwork in the sand.

"Sisyphus is a CNC machine," reveals Bruce. "It doesn't use G-code for its file format, but the principle is the same: a toolpath determines where the ball moves and its speed. With Sisbot being a polar machine, these moves end up producing spiral arcs, but this still works since small arcs connected together can emulate any path.

"My patterns tend to be algorithmic since I never learned to draw," Bruce continues. "But anyone can create paths for Sisyphus – just draw something without picking up your pen. If you can record the positions of your pen as you draw, you can compose for Sisyphus."

Above The Sisyphus table can contain more than one ball; they work together to create the artwork

Controlling Sisbot

A Raspberry Pi is the perfect computer to control the Sisbot and create the works of art, but it wasn't always that way. "For a very long time, all my motion-control artworks were controlled by Windows PCs running DOS," says Bruce. "In fact, three still do, running every day in their museums.

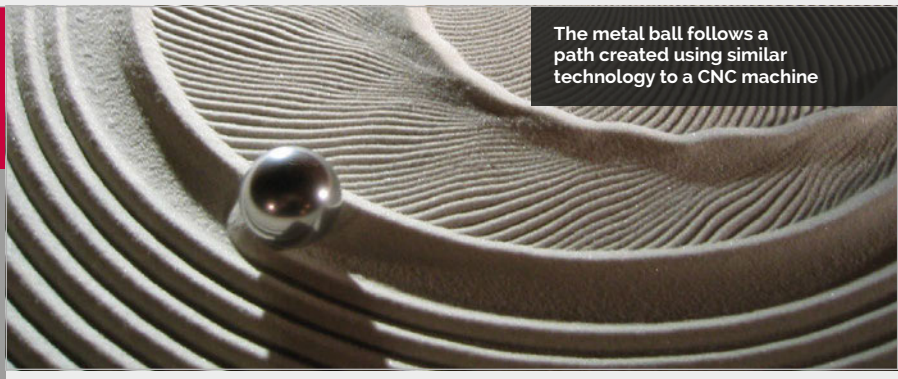
"I don't like change when it comes to something that works," Bruce admits. "Through many trials, and sometimes painful dead ends, I've learned that community matters."

> "I've been creating Sisyphus sculptures for nearly 20 years"

It was the Raspberry Pi community that convinced Bruce to switch over to a low-cost microcomputer. This was "more important than form factor and low cost," he tells us.

"The choice to use Raspberry Pi in the new home versions of

PRECISION BALL CONTROL ROBOT

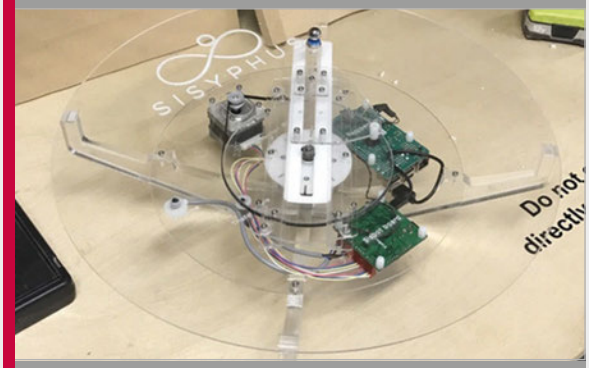


The metal ball follows a path created using similar technology to a CNC machine

>STEP-01

The Sisbot

Under the table is a two-motor robot called Sisbot. This moves a magnet which pulls the steel ball (sitting above the sand).



>STEP-02

Playing motion

The motors of the Sisyphus are controlled by a Raspberry Pi. This plays a set of path files, much like a music player plays an MP3 file.

>STEP-03

Always on

Sisyphus has no on/off switch. When it's plugged in, it automatically calibrates and starts playing. You connect to it with WiFi from a laptop or an iPhone app. From the app, you can control the speed of the Sisbot and the lighting of the table.

Sisyphus was driven by all the 30-somethings that I listen to," says Bruce. The community informed him that JavaScript is essential and Node.js runs well on a Raspberry Pi.

"The size of the community and its willingness to share makes the Raspberry Pi unique," adds Bruce.

The requirements of the Sisyphus table are complex. "Three levels are involved," he explains. At the lowest level is "firmware written in C, running on the SisBotBoard". This was created by Brian Schmalz (**magpi.cc/2fXxWhl**) for an earlier project Bruce created, called EggBot. "Brian started many years ago with the initial crude C code I wrote for running my steppers," says Bruce.

The second level is higher-level motion control, recently ported to JavaScript by Bruce and cleaned up considerably by Alex Wayne (**magpi.cc/2fXDqs7**). This code runs in Node on the Raspberry Pi.

The third level is algorithmic path generation. "I originally did this with AutoLISP routines running in vintage AutoCAD," says Bruce, "but now I use the Grasshopper Rhino 3D plug-in."

History matters

"I became fascinated with the challenge of connecting motors to my computer about 25 years ago," Bruce recalls. "The easiest and most compelling way to demonstrate its potential is to create a machine that can draw, using just two motors."

EggBot was Bruce's first art machine. "[EggBot] rescued me from being locked up as a madman and for being so obsessed with, and talking incessantly about, the potential of hooking up motors to a computer," he laughs.

"[EggBot] is actually pretty cool," he says, "and people I showed it to, for the most part, got that. I spent many years building successively larger drawing machines that eventually were capable of moving cutting tools like plasma torches to enable me to cut out intricate shapes, mostly in metal, to make sculptures.

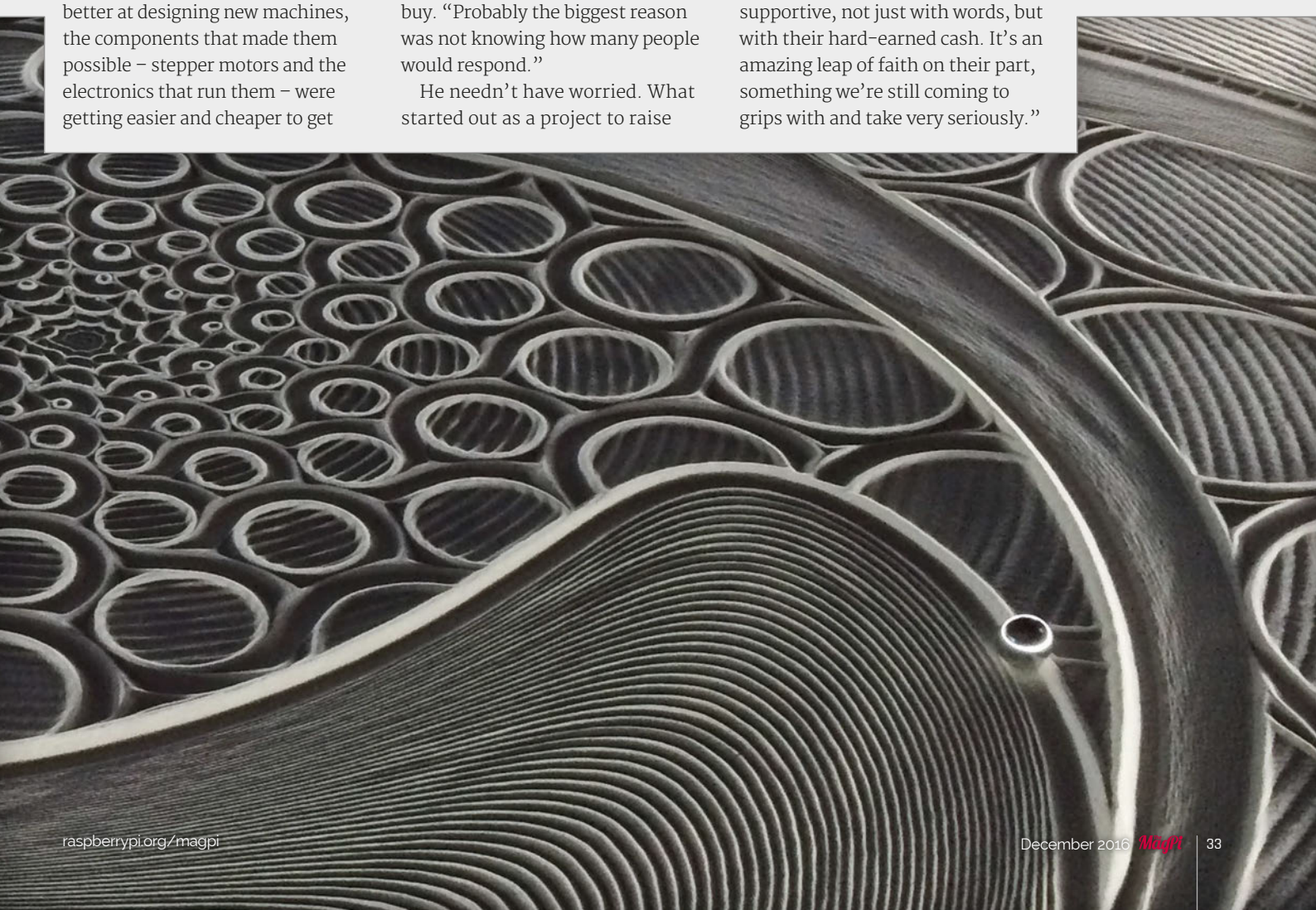
"I fell in love with watching them move," continues Bruce. "As I got better at designing new machines, the components that made them possible – stepper motors and the electronics that run them – were getting easier and cheaper to get hold of. In 1998, Sisyphus became my first CNC machine to escape the studio – transitioning from a tool used for making sculpture, to being the sculpture itself."

After 20 years of making, and refining, the Sisyphus project, Bruce was wary of making it available to buy. "Probably the biggest reason was not knowing how many people would respond."

He needn't have worried. What started out as a project to raise $50,000 eventually ended up getting close to two million dollars.

"When it comes to planning to make stuff for others, it helps to know how many will want it," says Bruce. "Kickstarter is a very public space to show what you want to do and find out how many people are supportive, not just with words, but with their hard-earned cash. It's an amazing leap of faith on their part, something we're still coming to grips with and take very seriously."

Above Users can control the lighting of the table, as well as the speed of the metal balls



MASATO SUDO

Masato Sudo is a principal software engineer in the Microsoft Windows IoT team. He builds IoT prototypes on Windows 10 IoT Core.
magpi.cc/2ePXY33

Faces are photographed using the webcam
The door is unlocked when the Raspberry Pi confirms the identity of a person
The Raspberry Pi is running Windows IoT Core. It uses Microsoft Face APIs within Project Oxford to detect registered users
Whitelisted Visitors
Live Camera Feed
Welcome to the Facial Recognition Door!

WINDOWS IOT: FACIAL RECOGNITION DOOR

Quick Facts

- It was built at Microsoft's OneWeek Hackathon event for employees
- It taps into the Face API from Microsoft Cognitive Services
- A Leviton 79A00 is used to lock and unlock the door
- It uses a regular Microsoft Lifecam HD3000
- C# was used to write the code

This Raspberry Pi-controlled door can spot a face, and open up for people it knows – strangers get shown the (closed) door

Microsoft has been busy making the most of the Raspberry Pi board with its Windows 10 IoT Core operating system.

Windows 10 IoT Core might not be the home version of Windows 10 that some users would like on their Raspberry Pi. However, it taps into some pretty powerful services and enables hackers to create unique projects.

"The Windows 10 IoT Core Facial Recognition Door is an example of a compelling new experience that can be created when Windows 10 IoT Core is combined with Microsoft Cognitive Services," says Masato Sudo, a software engineer in the Microsoft Windows IoT team.

He explains: "It's a prototype that identifies the visitor." If the door recognises the visitor, it will greet them by name, and the door will be unlocked and opened.

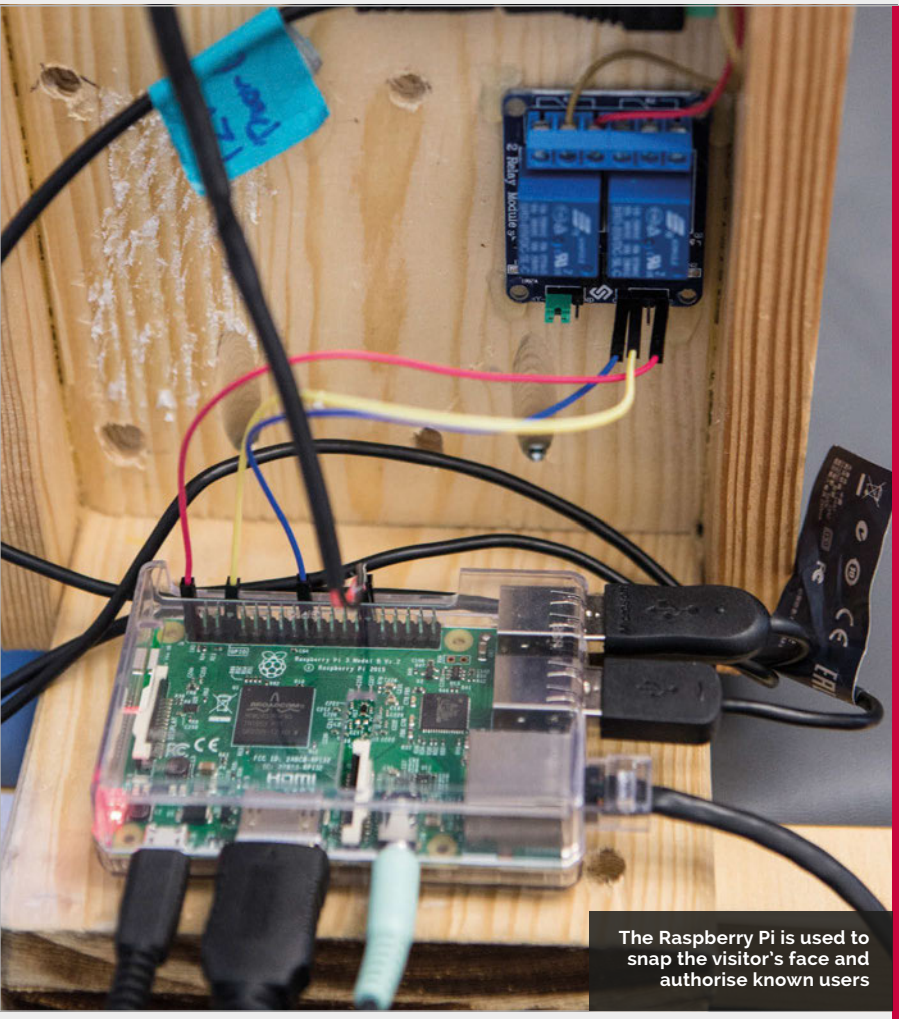
"If it doesn't recognise the visitor, it notifies the person that they are not recognised and keeps the door locked," Masato tells us. They're really shown the door.

Facial recognition is no mean feat, but the hardware requirements are surprisingly light. Attached to the Raspberry Pi is a standard USB camera to capture the face of prospective visitors. At the same time, a regular set of speakers is used to provide voice feedback to the guest.

For the door to open, a GPIO pin is wired up to a relay switch. When the Raspberry Pi sends a signal to the GPIO, the relay switch unlocks the door.

Finally, another GPIO pin is wired up to a doorbell. When the visitor presses the doorbell, it snaps a photo of their face, and the authentication process begins.

"Windows 10 IoT Core is the brains of the project," explains Masato. "For facial recognition, it uses a Microsoft Cognitive Services API that runs in the Microsoft Azure cloud."



The Raspberry Pi is used to snap the visitor's face and authorise known users

Microsoft Cognitive Services (**magpi.cc/2eDbDM6**) is the new name for Project Oxford. It "provides machine learning and artificial intelligence capabilities in the areas of vision, speech, language, knowledge, and search," explains Masato. For the project, he used the Face API (**magpi.cc/2eDb3xV**).

"We create a friends and family list, and add their pictures to a list in the Microsoft Azure cloud," says Masato. "We used the Face API's Face Grouping feature for this process. It's used to sort faces into groups based on their visual similarity. After a picture is taken of the visitor at the door, it verifies it's a human face before sending it for a face match."

If the image does contain a human face, it's sent to the Microsoft Azure cloud to detect if it's a match. "It searches matches of the face from the list we created," explains Masato, "and returns the confidence level of 0 to 100% on the likelihood of a face match.

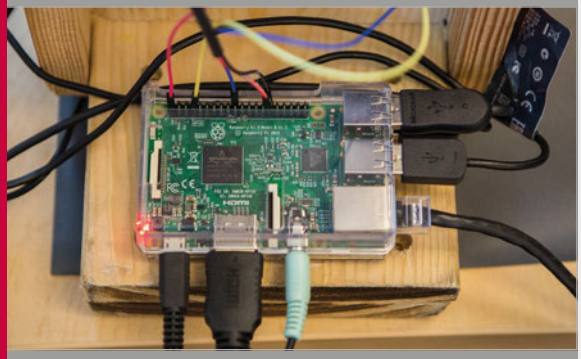
"We wrote our code in C# and used Visual Studio 2015 to code, deploy, and debug our project on Raspberry Pi," he continues. "We shared our code with the open-source community so that developers can enhance the security."

> "If it does not recognise the visitor, it notifies the person and keeps the door locked"

RECOGNISING FACES

>STEP-01
Smile for the camera

The visitor presses the doorbell, and a regular webcam takes a snap of their face. The speakers play an audio greeting to the visitor.



>STEP-02
Getting cognitive

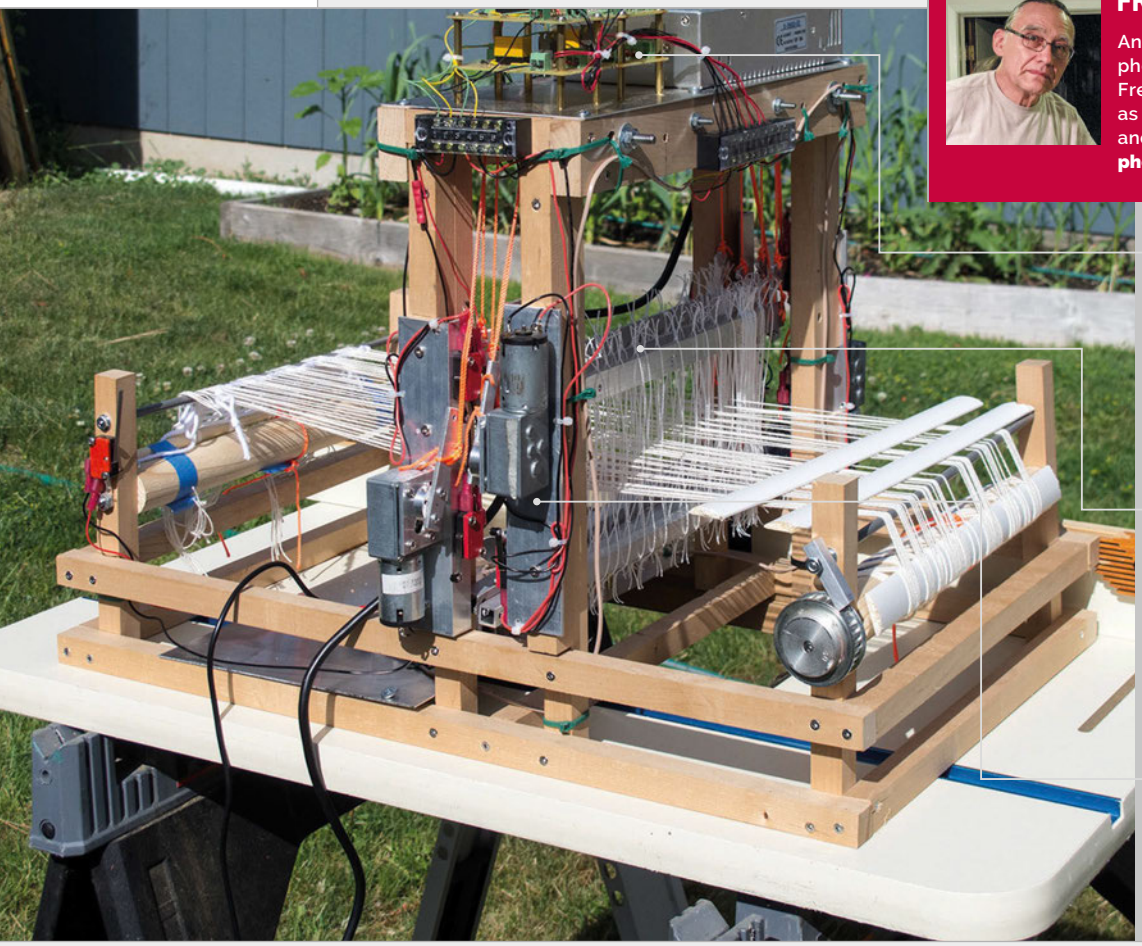
The Raspberry Pi is running Windows 10 IoT Core. It sends the photo to Microsoft Cognitive Services, where it's compared to a previous selection of authorised people.

>STEP-03
Welcome visitors

The visitor is matched against the authorised faces using Microsoft's Faces API. For positive matches, a relay is used to open the door, and the speakers welcome the visitor by name.

FRED HOEFLER

An award-winning landscape photographer from Spokane WA, USA, Fred enjoys weaving in his spare time, as well as tinkering with electronics and the Raspberry Pi.
photographic-perspectives.com



These custom-made motor control units feature 12V relays and TIP 120 Darlington transistors

Connected to an actuator arm by a cord, each loom harness is lifted and lowered in sequence

Controlled by the Raspberry Pi, four 12V DC motors are attached to actuator arms

Quick Facts

- The loom took Fred a year to build
- One Pi was accidentally fried in the process
- The total component cost is around $150
- Limit switches prevent the harnesses moving too far
- The user needs to send the weft threads through by hand

PI LOOM

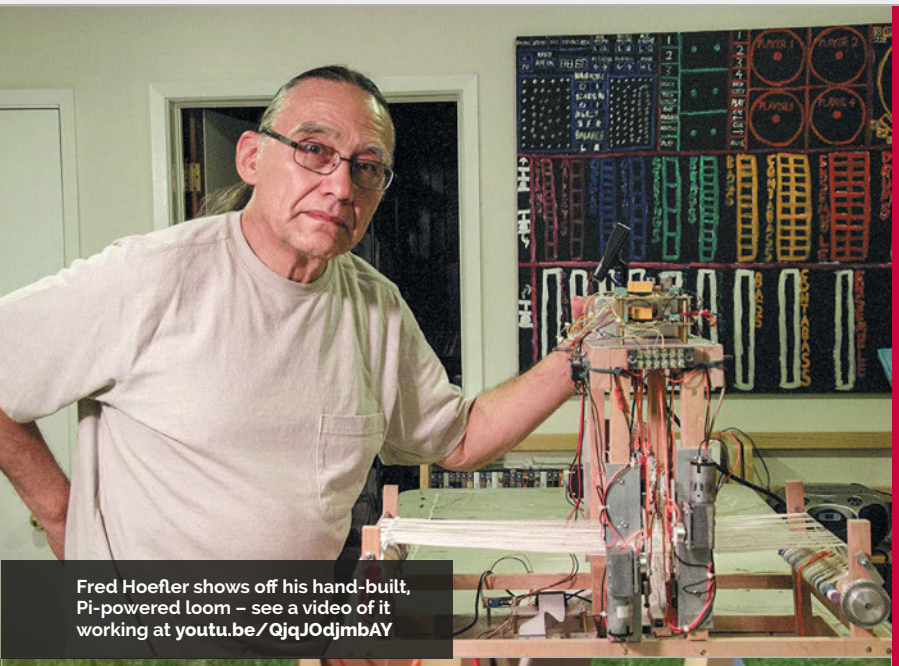
A weaving wonder controlled entirely by Raspberry Pi

When Fred Hoefler sent back a computer-controlled tabletop loom after it stopped working with his MacBook Pro, his wife Gina suggested that he should be able to build one with "one of those Raspberry Pi things of yours." Fred recalls, "The most reasonable answer I could come up with was: 'as a matter of fact, I think I can.'" A project was born (**magpi.cc/2fDminE**).

At the outset, Fred set out some ground rules for building the new four-harness tabletop loom. As well as being controllable by a Raspberry Pi via the Bash console, no part of the hand-built loom was to cost more than the Pi itself. It would also need to be quiet enough to carry on a conversation in the same room with it.

Following a year of work, with many stops and starts, the end result is a working Pi-powered loom that meets all Fred's goals. "Basically, the Raspberry Pi drives four 12-volt DC motors turning in opposite directions, like a robot that rolls forward and backward," he explains. "These motors are attached to actuator arms that are in turn attached to a cord that runs through a pulley system to individual loom harnesses."

Fortunately, Fred had some previous electronics experience from his earlier career as an aircraft mechanic and electrician. "I was quite used to working with wiring, relays, switches, and motors. Working with aircraft aluminium, making brackets and such is also second nature to me." The project wasn't without the odd mishap, however. As Fred discovered the hard way, "A Raspberry Pi cannot directly control common motor power. You must use some kind of motor control circuit. I thought I would try to use a half-bridge MOSFET controller (a TI SN754410) that claimed to be able to handle up to 1 amp, and have all sorts of protection between the power circuit and the GPIO circuit." After one working motor test cycle, the result was a fried Pi: "We held an appropriate memorial."



Fred Hoefler shows off his hand-built, Pi-powered loom – see a video of it working at **youtu.be/QjqJOdjmbAY**

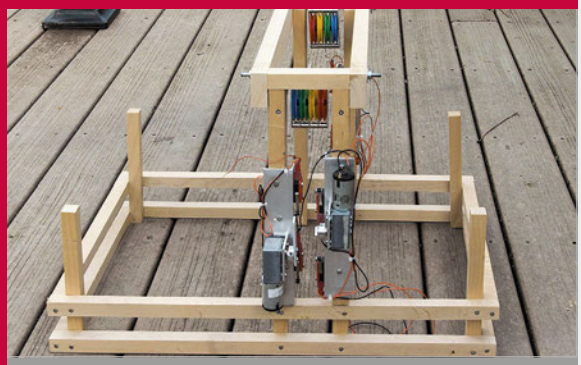
Each 12V motor moves an actuator arm with a cord attached via a pulley to a loom harness

To his surprise, Fred had less trouble programming a Python script (which can be found on his website) to control the loom. "Python works like the old BASIC interpreters we used back in the early 1970s, only easier. There are lots of Python examples out there explaining how to turn on and off GPIO pins... Once I set up how to control one harness with a module, all I had to do was copy-paste a duplicate module and change the calling parameters to control the next harness."

> "Four 12-volt DC motors turn in opposite directions, like a robot that rolls forward and backward"

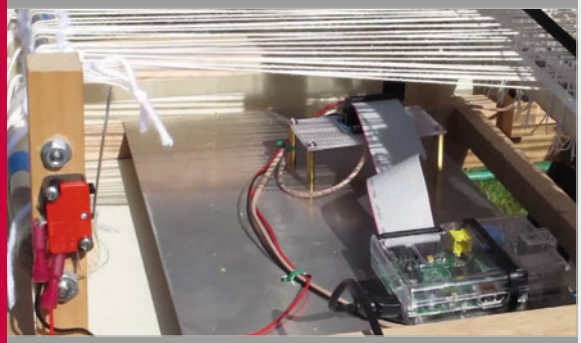
While Fred admits the loom is too slow to use for production purposes and requires an operator to send the weft threads through the shed by hand, he is looking to improve it by experimenting with different 12V actuators and possibly solar power. "Finally, I found that the manufacturer of the 'cute little loom' that started this whole project has halted production due a supplier not being able to produce a reliable control unit. If we could just send him a copy of this article and offer to upgrade the control unit..."

BUILD A PI-POWERED LOOM



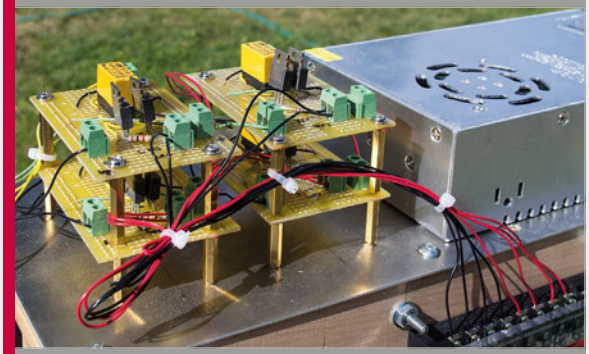
>STEP-01
Frame and motors

Attached to the hand-built wooden frame with metal brackets, the four 12V motors are linked to actuator arms, with cords running via a pulley system to lift and lower the harnesses.



>STEP-02
Pi powered

A Raspberry Pi is connected via a breakout board to the rest of the circuitry in the loom. Since the Pi cannot directly control 12V DC current, motor control units are required...



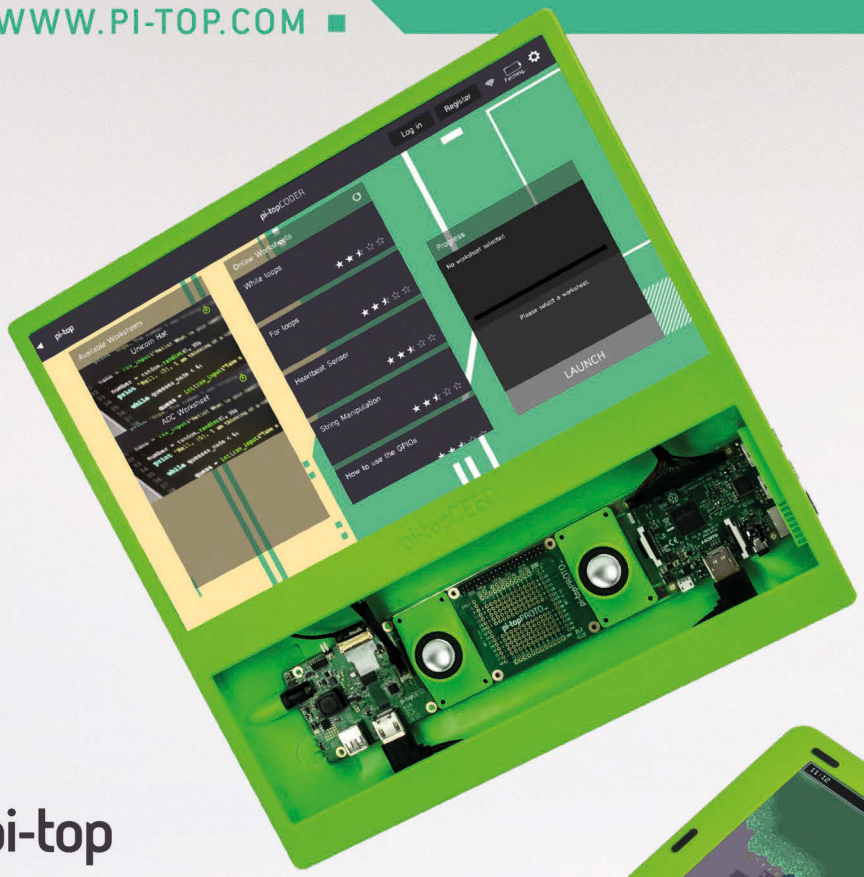
>STEP-03
Motor control units

Fred opted to build his own motor control units using 12V relays and TIP 120 Darlington transistors. An alternative would be to use a commercially available Pi relay control board.

pi-top

LEARN
PLAY
CREATE

WWW.PI-TOP.COM



pi-topCEED

- Adjustable Viewing Angles
- 14" HD Screen
- Modular Components

$114.99
without Raspberry Pi
ex VAT

pi-top

- 10 Hour Battery Life
- 13.3" HD Screen
- Modular Components

$264.99
without Raspberry Pi
ex VAT

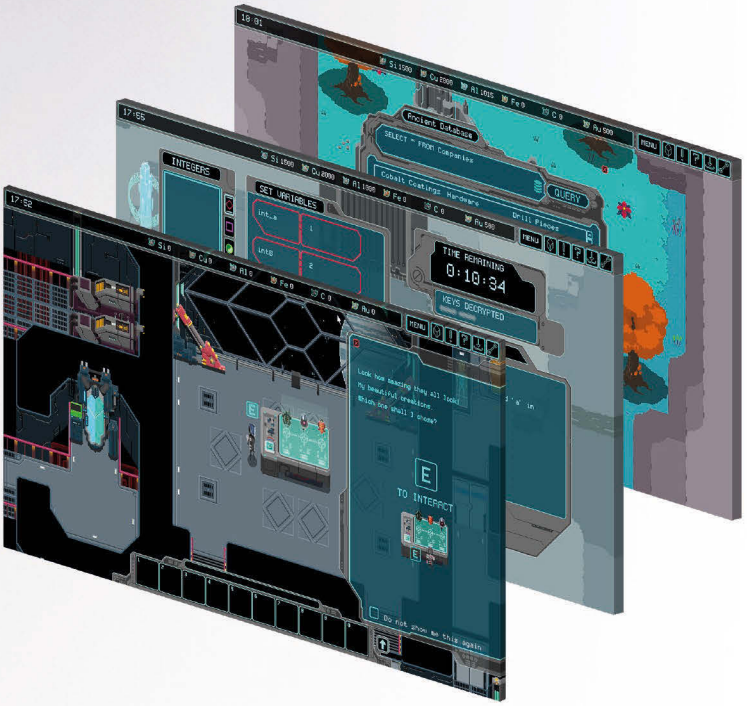
Available in green or grey colours

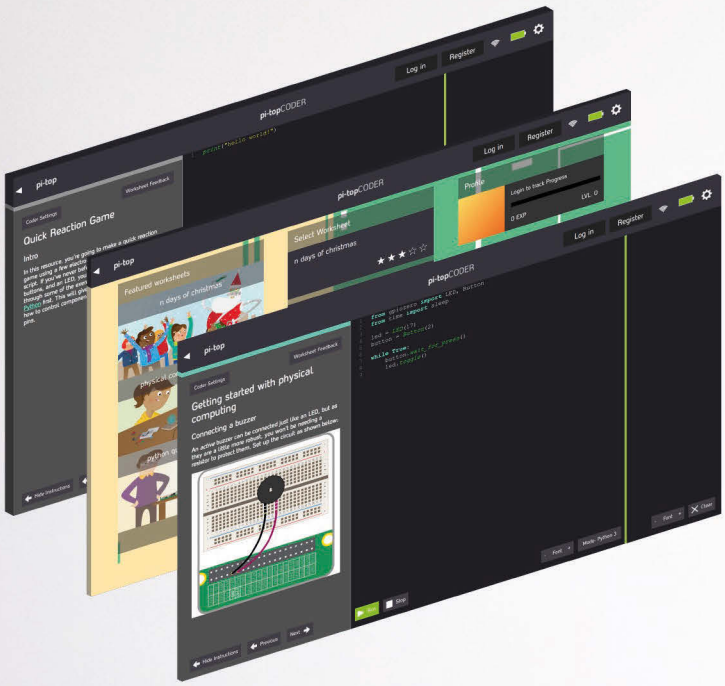
pi-top | Worldwide shipping available in green or grey at www.pi-top.com
Stay up to date with our latest news by following our social media

CEED**universe**

CEED**universe** is a world of fantasy grounded in computing reality! After crash landing on a strange new planet you will first encounter 'drag and drop' coding puzzles that improve your computational thinking skills.

Discover more about the planet you've landed on and the civilisation that used to inhabit it while learning harder and more complex concepts in fun and engaging ways! Before long you'll be writing your own conditional statements, loops and functions. Find out more about CEED**universe** at www.pi-top.com

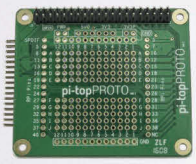




pi-topCODER

Influenced by the workflow of makers and hackers, pi-top presents **pi-top**CODER - an integrated code editor which allows you to learn, write and test code all in one view. With intuitive syntax highlighting, dynamic views and customizable themes it makes for a versatile learning tool for your projects.

pi-topCODER also has every Raspberry Pi Foundation lesson plan created and will track and save your progress as you go through dozens of fun hardware and software projects.



pi-topPROTO

pi-topPROTO is a HAT compatible Add-on Board for your **pi-top** or **pi-top**CEED that allows you to prototype electronics. Create a Weather Station, HAM Radio, Heart Rate Monitor, or integrate any Arduino based maker kits into your own Raspberry Pi compatible prototyping board!

pi-topSPEAKER

Give a voice to your **pi-top** device with **pi-top**SPEAKER!

- Modular design, attach up to three in a row to give true stereo sound.
- 2W per module
- Left, Right and Mono mix selection
- High quality SPDIF digital audio from HDMI
- I²C controlled

Our Official Distributors
adafruit
HANDS-ON SCIENCE
KIWI electronics
KUBII
MODMYPI
MICRO CENTER computers & electronics
Pi-Shop.ch
RS
sparkfun ELECTRONICS

HOW TO USE A BREADBOARD

This humble plastic block full of holes can be used to create just about anything

You'll Need

- Breadboard
- LED light
- Resistor
- Male-to-female jumper leads
- Male-to-male jumper leads

Most of our projects are tested using a small piece of plastic known as a breadboard. Officially, it's known as a 'solderless breadboard' because it enables you to use circuit parts without soldering them together.

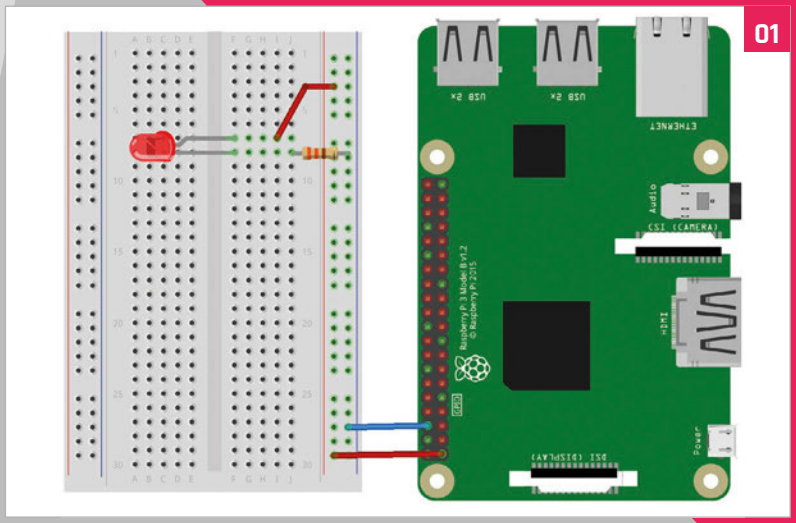
Electrical components are connected by pushing them into the holes in a breadboard. These holes are connected in strips, as shown in the main image. If you push a wire, or a different component, into one hole in a strip, and another wire into the hole next to it, it's as if you'd physically joined (or soldered) the two wires.

In the old days, people would either solder wire components together on an actual breadboard, or they'd wrap wires together around nails in a pinboard.

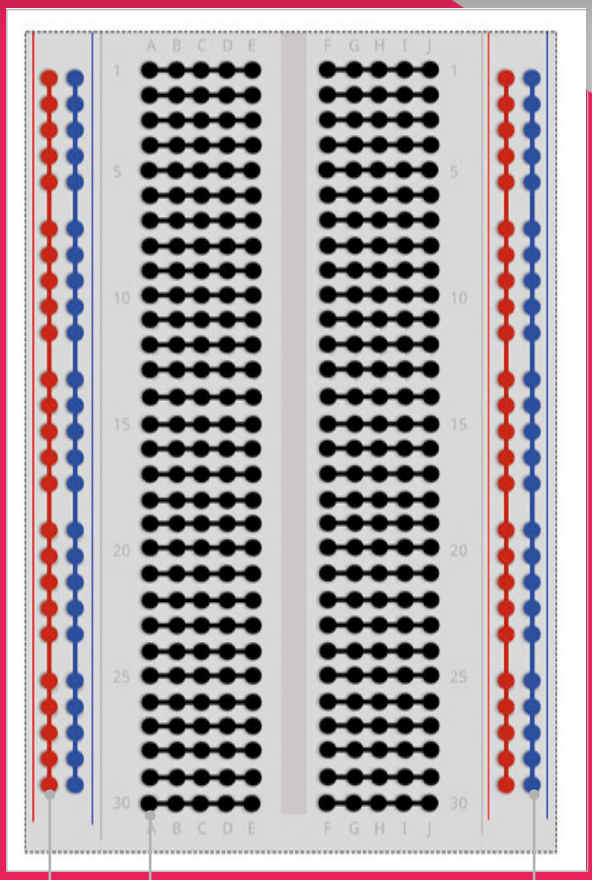
For a lot of Raspberry Pi fans, using a breadboard is part of life. But for many newcomers this quirky piece of kit is baffling: a smorgasbord of holes arranged in rows and columns that seem to make little sense.

So we think it's high time we had a beginner's guide to how a breadboard works. In this tutorial, we'll explain how these holes are arranged, and how to set up a circuit on your breadboard.

If you already know all this, feel free to move on. If not, stick around and learn about one of the most fun things you can do: building your own circuits and hooking hardware up to your Raspberry Pi.



01



A hole in the blue (sometimes black) rail is connected to a GND pin. It then becomes the ground rail

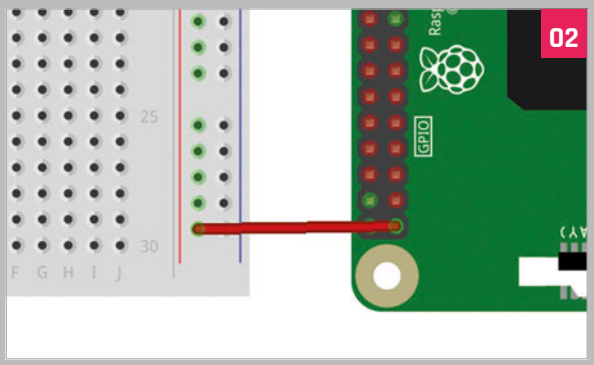
These horizontal rows are where you place the components that make up your circuit. They are linked into groups of five (or six) holes

These long vertical strips are known as rails. They are used to provide regular constant power. The red rail is connected to a power pin and becomes the live rail

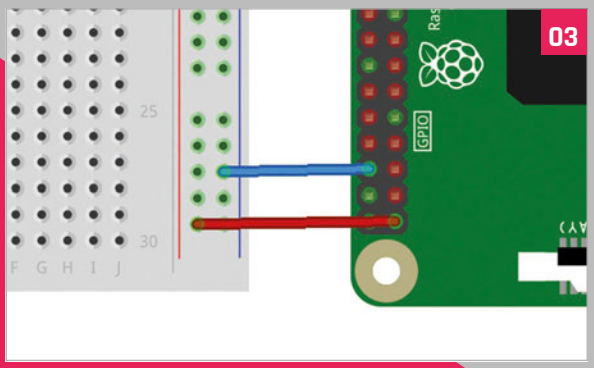
>STEP-01

Fritzing diagram

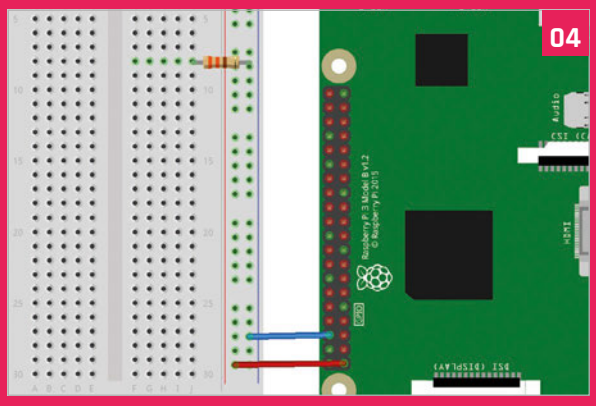
Circuit diagrams can be a little hard to understand for the novice. So we use visual breadboard diagrams, like this. This complete diagram uses the power and ground pins from a Raspberry Pi to light up an LED.



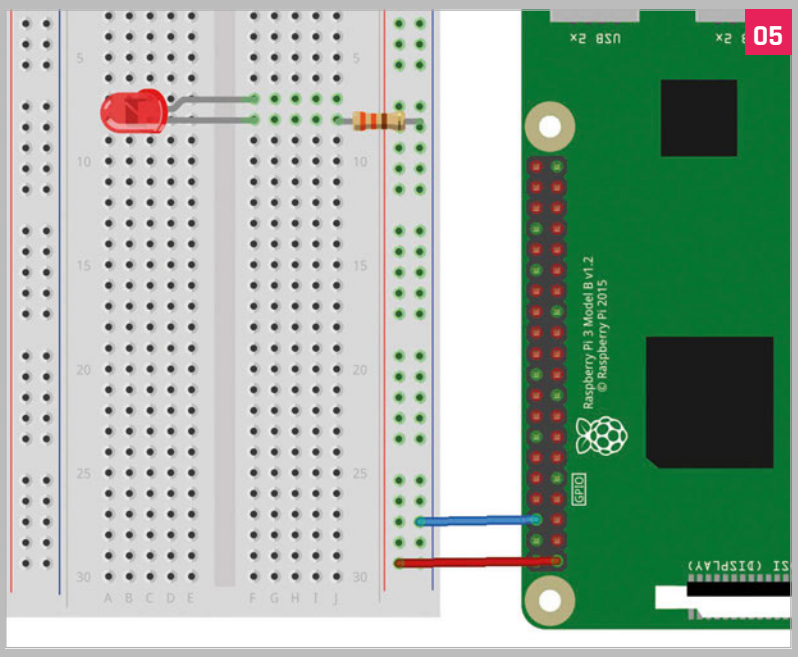
>STEP-02
Wire up the live rail
Take a female-to-male jumper lead (the colour of the wire doesn't matter) and connect the female end to a 5V pin on the Raspberry Pi. Place the male end of the lead into a hole on the red rail on the breadboard.



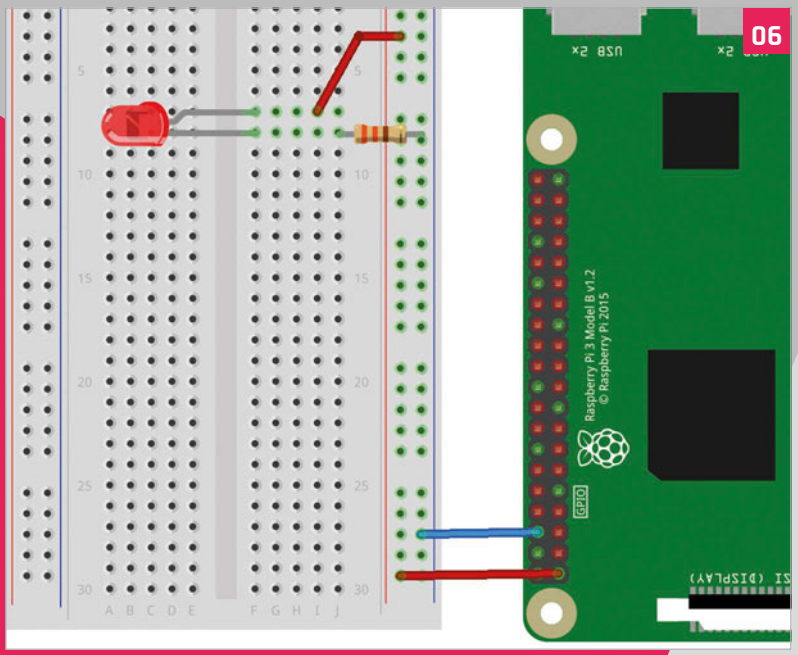
>STEP-03
The ground rail
Take another female-to-male jumper and connect the female end to a ground (GND) pin on the Raspberry Pi. The male end goes into a hole on the blue (ground) rail. All blue holes now act as a ground pin.



>STEP-04
Add a resistor
Take a resistor and connect one leg of it to a hole on the ground rail of the breadboard. It's now linked to the ground pin of the Raspberry Pi (via the jumper lead we used in the previous step). Take the other leg and connect it to a hole on the main breadboard.



>STEP-05
Add the LED
Take an LED component and look at the legs. Notice that one of the legs is shorter than the other. Place the shorter leg in a hole on the same row as the resistor. This leg is now connected to the resistor (which is linked to the ground rail, and therefore to the ground pin on the Raspberry Pi).



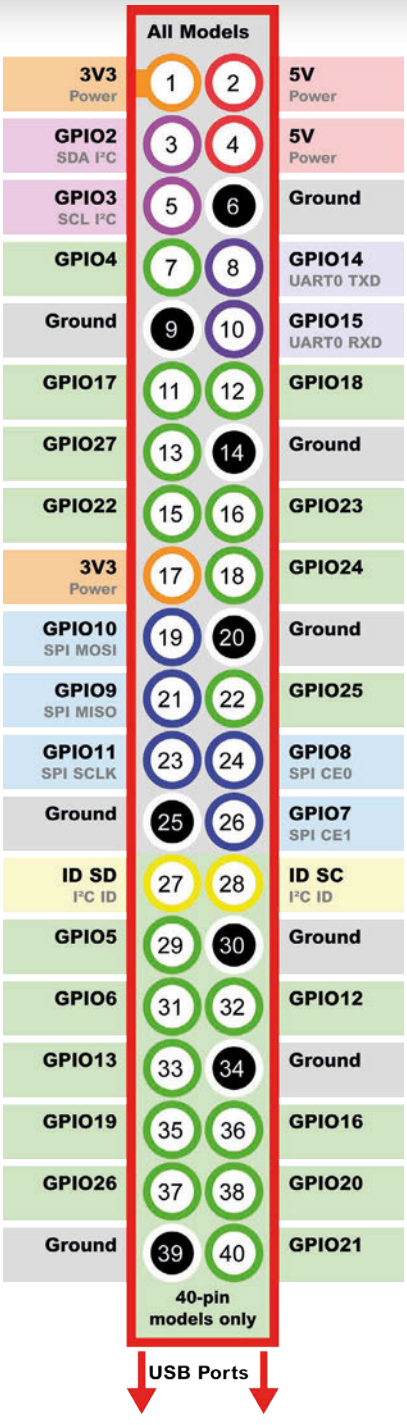
>STEP-06
Wire it up
Place the longer leg in a hole on the next row along. Now take another male-to-male jumper lead and place one end in the hole next to the long leg of the LED. Place the other end in a hole on the red live rail to complete the circuit. The LED lights up.

BEGINNER'S GUIDE TO GPIO ZERO

Discover the easy way to control GPIO pins on your Raspberry Pi

GPIO Zero Layout Guide

GPIO Zero uses the BCM number system to identify pins. These are the numbers referenced on the outside of the diagram, such as 18 for GPIO18. The board numbers (the ones on the inside) are just to help you count the pins. You can use any pin here marked GPIO, but the ones in blue double up with other functions. For this reason, we will stick to the ones marked in green in this tutorial.



You'll Need

- Rasbperry Pi
- Breadboard
- LEDs
- Button
- Jumper cables

The 40 metal pins on your Raspberry Pi board are known as General Purpose Input/Output pins, or GPIO pins for short.

These pins are your connection between the virtual world of computer code and the real world. With GPIO, you can connect circuit components up to your Raspberry Pi.

It's easy to attach LED lights, buttons, buzzers, and all kinds of electronic components to your Raspberry Pi. Typically you connect these components to a breadboard, at least when you are starting out. A breadboard is a plastic prototyping board used to link circuit components together without having to connect them physically. The legs of the components, along with cables, are pushed into adjoining holes on the breadboard.

You then hook the breadboard up to the GPIO pins on a Raspberry Pi (again, using jumper cables). You'll need male-to-male jumper cables to connect breadboard components together, and male-to-female jumper cables to connect the breadboard to the GPIO pins on your Raspberry Pi.

Different GPIO pins have different qualities. Some provide constant power, either at 5 volts or 3.3 volts. Others are ground pins, which should be used to complete a circuit.

Perhaps the most interesting GPIO pins are the ones that can be programmed. These can be turned on and off, powering up components (like LED lights) from code. Alternatively, they can be set to respond to voltage change inputs, such as a button push. Your code can then respond to these inputs.

Typically, you'll program GPIO pins using Python, although Scratch, Java, and a host of other languages can be used to control GPIO.

Programming GPIO in Python used to be quite a detailed task. But a new library, called GPIO Zero, simplifies things massively. With GPIO Zero you can quickly connect components and start using them with just a few simple commands.

>STEP-01
Set up a circuit

Programming GPIO pins is a relatively straightforward process, but remember that there are a few steps required to perform the most basic of tasks.

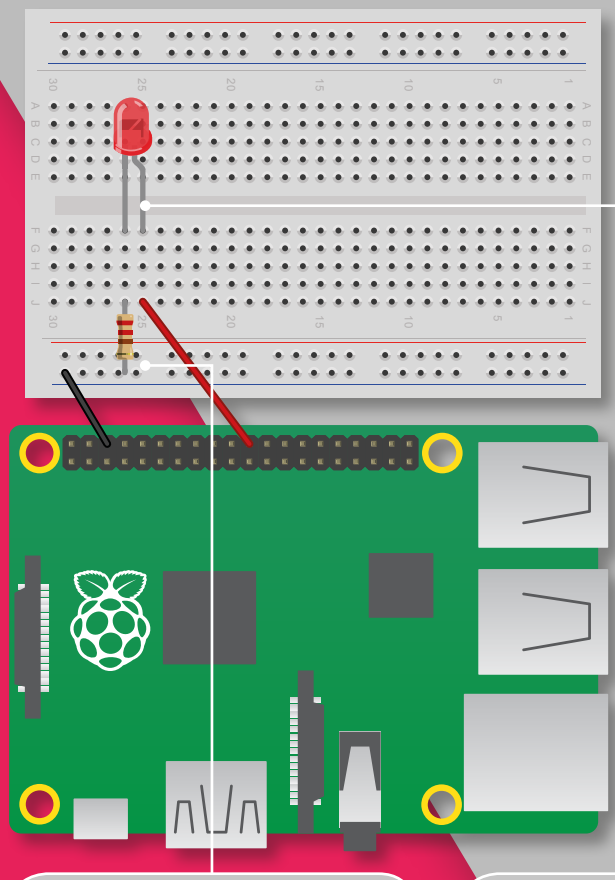
We're going to start by wiring up a single LED to GPIO25 as shown in this diagram. If you don't know how to use a breadboard, take a look back a couple of pages in this magazine. This circuit is extremely similar to the one we used there, except that instead of the long leg of the LED being connected to a live pin, it is connected to a programmable GPIO pin. This makes the LED light up when the GPIO pin is turned on in our program.

Use a female-to-male jumper lead to connect the GND pin to the ground rail on the breadboard.

Insert one end of a resistor into a hole on the ground rail, and the other end into a hole on the breadboard.

Connect the short leg of the LED into a hole on the same line and the longer pin on the line next to it.

Finally, take another male-to-female jumper wire. Place one end in a hole on the same row as the longer LED leg. Connect the other end to GPIO18.



A resistor is used to prevent the LED from burning itself out. The smaller the resistor, the brighter the light will be, but don't burn your LED out. A 330Ω resistor is a good start, though you can use other sizes. However, if you pick one that's too high, you may not be able to see the LED

The LED has two legs. The short one is connected to the resistor (and to the ground pin). The long leg is connected to GPIO25 on the Raspberry Pi

>STEP-02
Regular GPIO

If we were going to light up this LED using regular code rather than GPIO Zero, this is what we would have to write (don't enter this code: it's just an example):

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(25, GPIO.OUT)
GPIO.output(25,GPIO.HIGH)
```

It's not impossible to decipher. But this code is fussy, and concepts like 'OUT' and 'HIGH' get in the way of understanding the relationship between the code and the light. With GPIO Zero it's a lot easier.

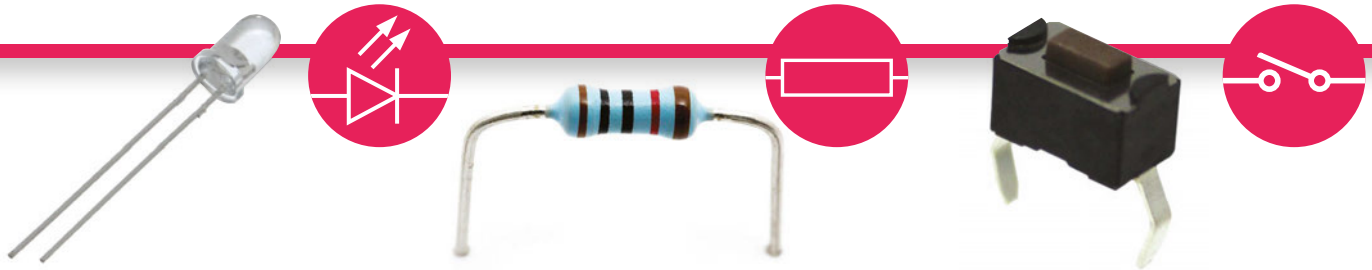
>STEP-03
GPIO Zero

GPIO Zero simplifies things. The same code in GPIO Zero looks like this:

```
from gpiozero import LED
led = LED(25)
led.on()
```

Enter each line of the code above into the Python shell one line at a time. Press **RETURN** after each line. When you enter `led.on()` the LED will light up.

As well as consisting of fewer lines, this program is easier for young students to understand.



LED
LEDs are easy to start with. LEDs have one leg longer than the other. They only work one way around.

RESISTOR
Always use a resistor with LED lights. This prevents them from burning out from high voltage.

BUTTON SWITCH
A button switch completes a circuit connection when it's pressed. Code can respond to button pushes.

The first line of the program is where we import the **gpiozero** library into Python.

The second line creates a LED object, with the pin number as the argument (inside the brackets).

The third line tells the pin to switch on.

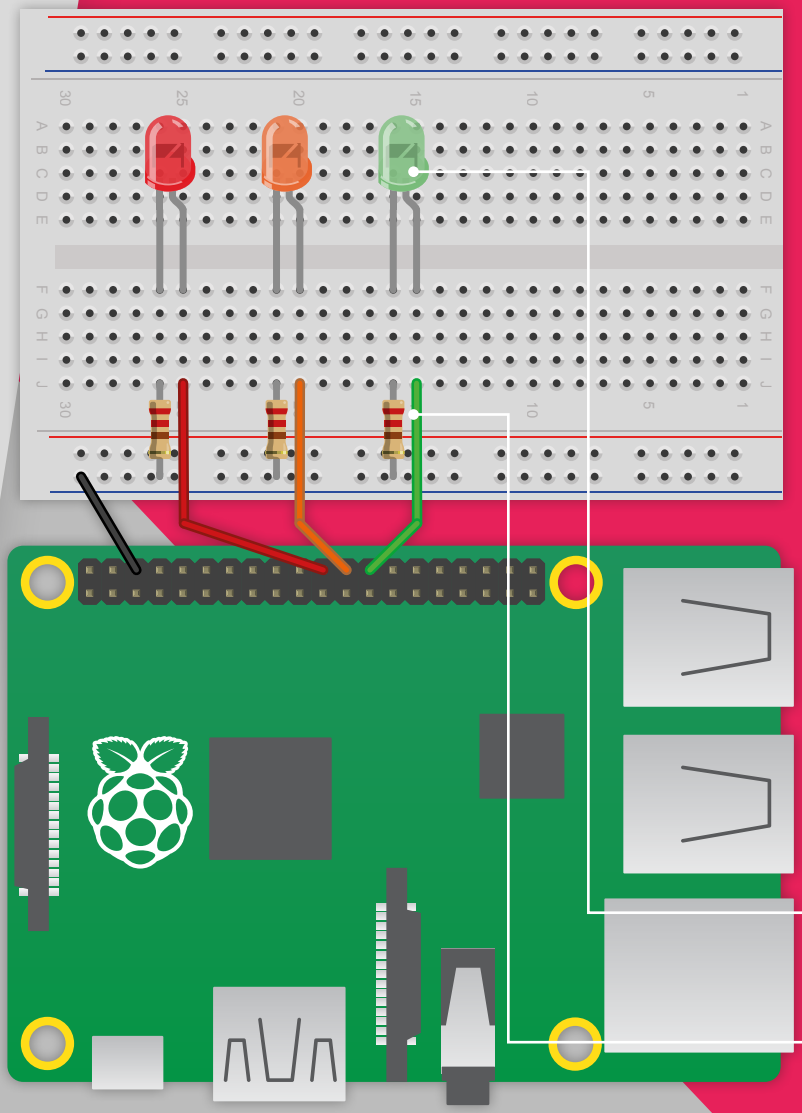
Now enter **led.off()**. The light will switch off.

>STEP-04
Blink

The real joy of GPIO Zero is that it lets you perform rather complex tasks with simple instructions. Creating a blinking LED with more standard code requires you to import the **time** module, set an LED to on, pause for a period, then set the LED to off, and repeat in a loop. This process makes it difficult to perform other code tasks at the same time. In GPIO Zero, however, you simply enter this line:

```
led.blink()
```

The light will start blinking on and off at one-second intervals. Enter **led.off()** to stop it.



The short leg of each LED is connected to a resistor. Each longer LED is connected a separate GPIO pin. These pins are used to turn on, or off, the power to each light

All three resistors are connected to the ground rail. This rail is connected to a single GND pin on the Raspberry Pi

>STEP-05
More control

One neat thing about GPIO Zero commands is that you can enter arguments inside the brackets. Enter **led.blink(** and stop at the open bracket. A yellow box appears showing the text '**on_time=1, off_time=1, n=none, background=True)**'.

These are the parameters available for the **blink** method. They are the number of seconds for which a light stays on and then stays off, how many times the light blinks, and whether you can add more code while the light is blinking.

After each argument is the default value: one second on, one second off, none (which means the light blinks until you say otherwise), and True (which lets you carry on adding code while the light is flashing). To set default values, add a number for each value (from left to right):

```
led.blink(4,2)
```

The light comes on for four seconds, and off for two seconds. Enter **led.off()** to stop it. Alternatively, you can add the item and equals sign to pick a value to change (and keep the defaults).

```
led.blink(n=3)
```

The light will blink three times and stop.

>STEP-06
Traffic lights

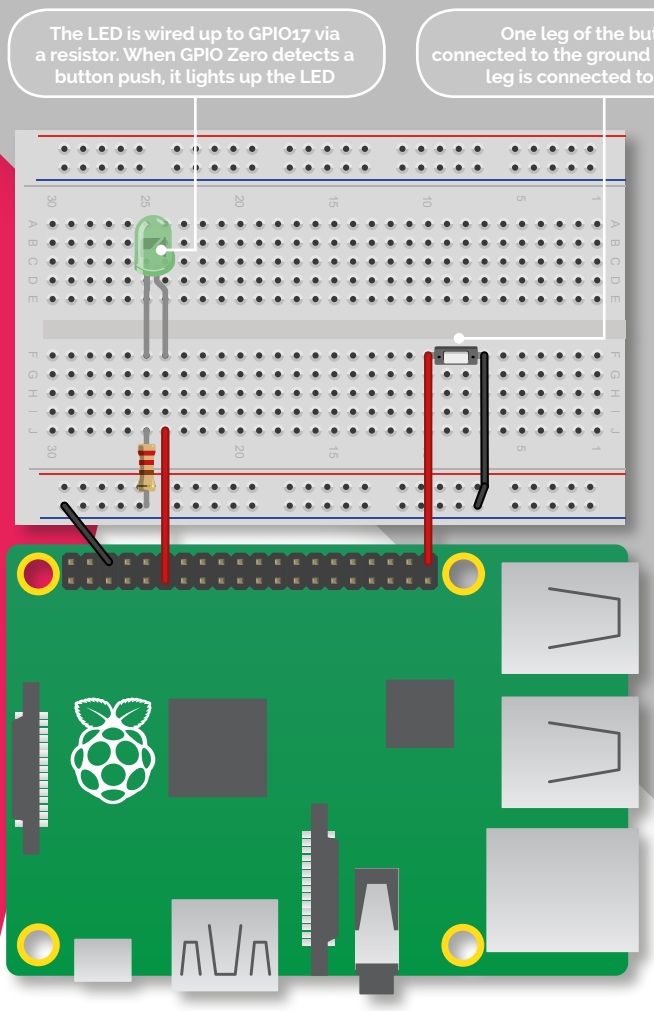
Let's take our LEDs and build something a little more complex. We're going to add another two LEDs to our circuit (three in total). We'll use one red LED, one amber, and one green.

Connect the new LEDs into the circuit using another two resistors to connect their shorter legs to the ground rail.

Connect the longer leg of the amber LED to GPIO8 and the longer leg of the green LED to GPIO7. These are the two pins next to GPIO25, so you have your LEDs all together.

Now we need to create a program to control our traffic lights. Open Python 3 (IDLE) and choose **File > New File**.

Type out the code from **traffic_lights.py** and choose **Run > Run Module** (or press **F5**). The Python program will run, and you'll see your traffic lights in action.


The LED is wired up to GPIO17 via a resistor. When GPIO Zero detects a button push, it lights up the LED
One leg of the button is connected to the ground rail. The other leg is connected to GPIO21

>STEP-07

Adding a button

Now we're going to add a button to our circuit and connect it to GPIO21 with an LED wired up to GPIO17. Buttons are wired up in a similar fashion to LEDs, using female-to-male jumper leads.

Like LEDs, buttons have legs. One leg is wired to a GND pin (via the same ground rail you've been using for LEDs); the other leg is connected to a GPIO pin. Some buttons have four legs, so you can connect them to the breadboard with their legs straddling the central groove (see the breadboard diagram).

>STEP-08

Button responder

Unlike LEDs, you don't need to use a resistor with a button. The button's legs are the same length, and it typically doesn't matter which way around you hook it up to the breadboard.

When the button is pushed, it forms a connection between the ground rail and the pin, completing a circuit. The Raspberry Pi is set up, using GPIO Zero, to detect this connection and respond.

Open Python shell and create a new file. Enter the code from **button.py**, save the file, and run the code. When you push the button, the LED will light up.

Language
>PYTHON

DOWNLOAD:
magpi.cc/2ehTqVq

button.py

```
from gpiozero import LED, Button
from signal import pause

led = LED(17)
button = Button(21)

button.when_pressed = led.on
button.when_released = led.off

pause()
```

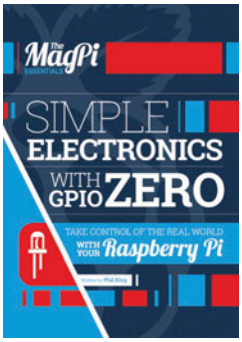
traffic_lights.py

```
from gpiozero import LED
from time import sleep

red = LED(25)
amber = LED(8)
green = LED(7)

green.on()
amber.off()
red.off()

while True:
    sleep(10)
    green.off()
    amber.on()
    sleep(1)
    amber.off()
    red.on()
    sleep(10)
    amber.on()
    sleep(1)
    green.on()
    amber.off()
    red.off()
```

Going further

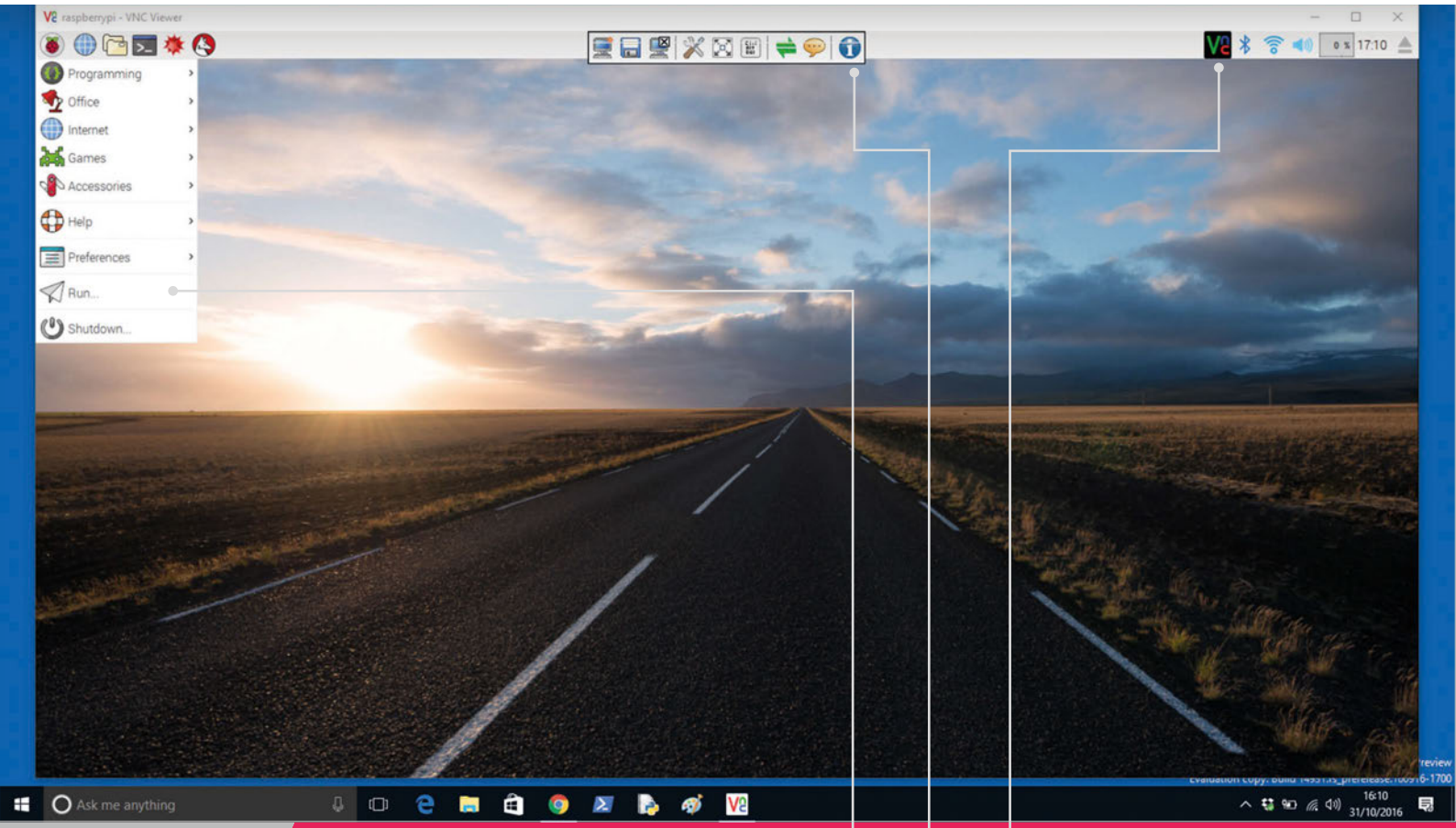
You can achieve a lot more with GPIO Zero than controlling LEDs and buttons.

For a more detailed look at how you can detect motion, control robots, read sensor information (such as movement sensors or thermometers), *The MagPi*'s Phil King has created a fantastic guide: *MagPi Essentials: Simple Electronics with GPIO Zero*. Learn more at **magpi.cc/Back-issues**.



BEGINNER'S GUIDE TO VNC

Remotely control your Pi from another computer with VNC Server and Viewer



You'll Need

- Raspberry Pi
- Raspbian
- VNC Viewer on your computer, smartphone or tablet device

VNC (Virtual Network Computing) is a great technology included with Raspbian.

With VNC, you can remotely control your Raspberry Pi from another computer, such as a PC or Mac, or even another Raspberry Pi board.

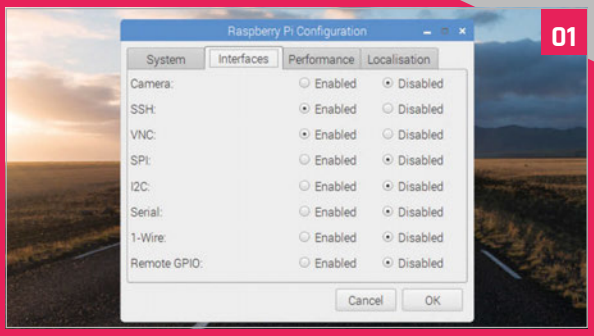
Sometimes it's not convenient to work directly on a Raspberry Pi. This can be because it's not easy to access, or because your keyboard and monitor are being used on your main computer.

With VNC, you can open the Raspbian desktop interface from your Raspberry Pi inside a window on your computer. You can even use VNC apps on your smartphone or tablet to control your Raspberry Pi.

By default, VNC Server from RealVNC gives you direct control over your Raspberry Pi, just as though you were sitting in front of it.

This is great for controlling lightweight Raspbian projects, such as IoT builds.

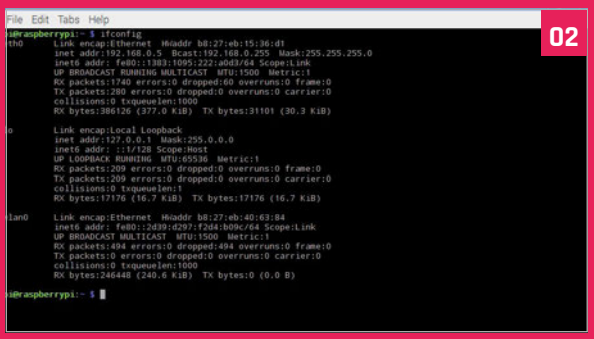
As we'll see in the steps, Raspbian Jessie with PIXEL includes VNC Server by default. However, you will need to enable it yourself. From then on, VNC Server will be loaded every time you switch on your Raspberry Pi.



>STEP-01

Enable VNC

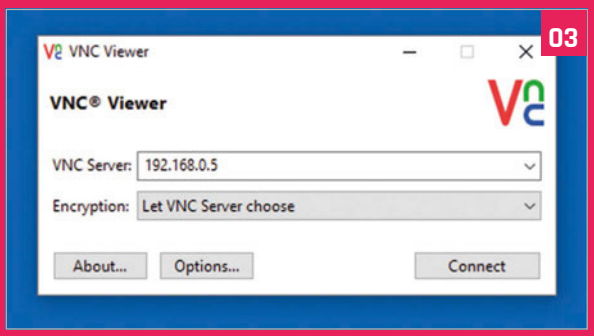
Start with a fresh build of Raspbian Jessie with PIXEL. Choose **Menu > Preferences > Raspberry Pi Configuration**. Click on Interfaces and set the VNC option to Enabled. Click OK.



>STEP-02

Network

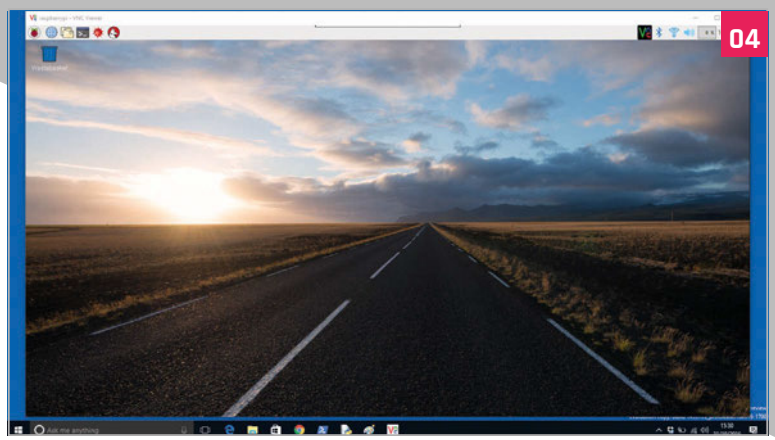
You'll need to know which IP address your Raspberry Pi is using to connect to it remotely. Open a Terminal window and enter **`ifconfig`**. Check for the four numbers next to **`inet addr`**. They'll be under **`eth0`** if you're connected using an Ethernet cable, or **`wlan0`** if you've connected to a wireless network. It'll look a bit like this: 192.168.0.5.



>STEP-03

Download VNC Viewer

Download and install VNC Viewer on your computer from RealVNC (**magpi.cc/1M4uzfG**). Open the app and enter the IP address from the previous step into the VNC Server field. Ensure Encryption is set to 'Let VNC Server choose', and click Connect then OK. The first time you connect, it will display a 'VNC Server not recognized' alert. Click Continue.



>STEP-04

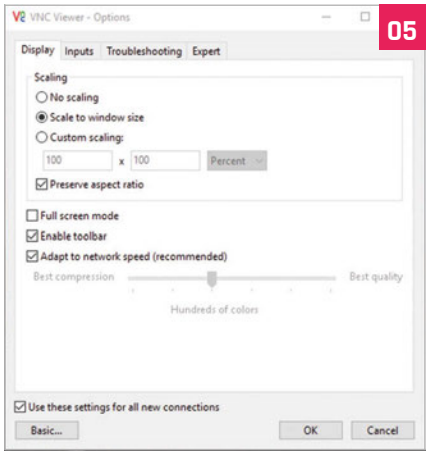
Desktop window

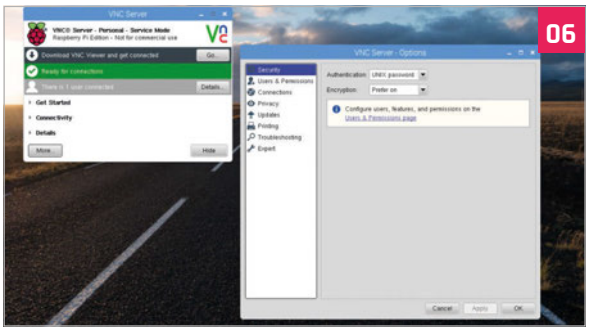
Enter **pi** into the Username field, and **raspberry** (or your Raspbian password) in the Password field. Click OK. The Raspbian PIXEL interface will appear inside a window on your computer. You can click on the Menu, open programs, and run Terminal commands on your Raspberry Pi. If you have your Pi connected to a monitor, you'll see it move as you remotely control it.

>STEP-05

Toolbar settings

At the top of the virtual window is a small white strip. Hover the pointer over it to reveal a set of control icons known as the Toolbar. Full-Screen Mode is one of the most useful. Click Settings to the left to access options; the Advanced button near the bottom expands the list of settings.



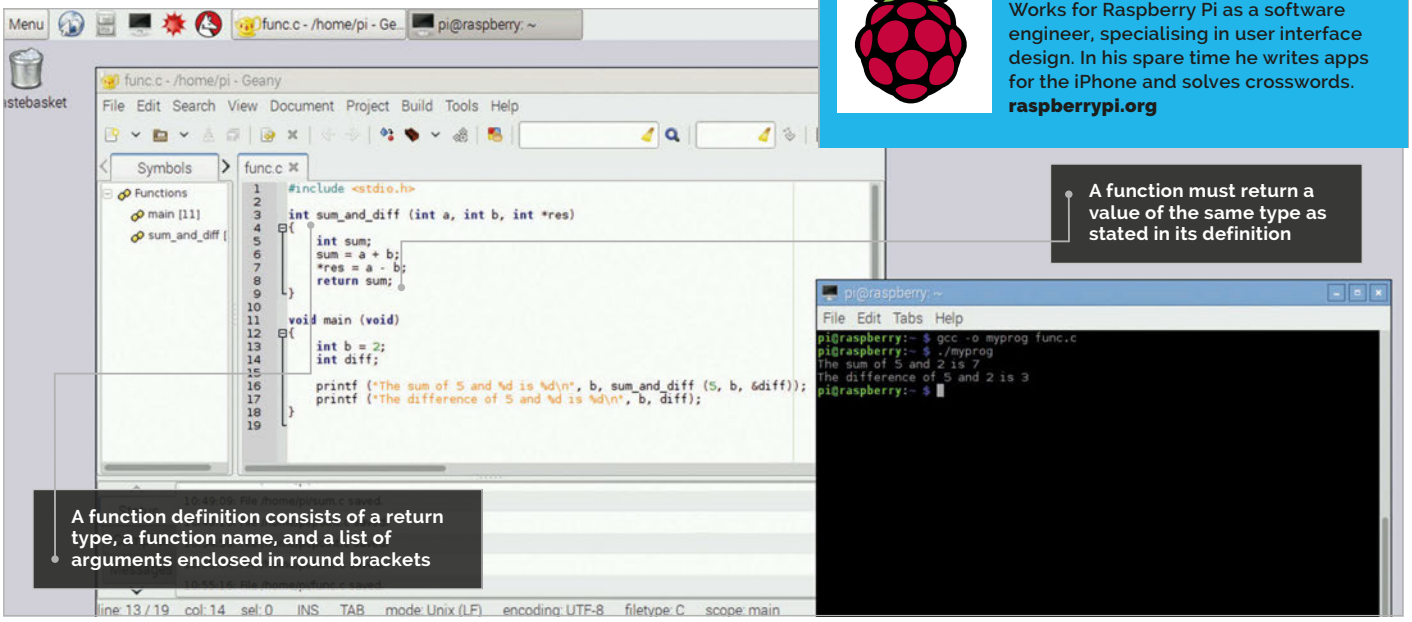


>STEP-06

VNC Server

Another set of options is found inside the VNC Server app on your Raspberry Pi. Click on the VNC Server icon to view the server window. Click on Details to see which users are connected (normally this will be just one). Click **More > Option** to view detailed settings for the VNC Server software.

SIMON LONG
Works for Raspberry Pi as a software engineer, specialising in user interface design. In his spare time he writes apps for the iPhone and solves crosswords.
raspberrypi.org


A function must return a value of the same type as stated in its definition
A function definition consists of a return type, a function name, and a list of arguments enclosed in round brackets

AN INTRODUCTION TO C PART 06

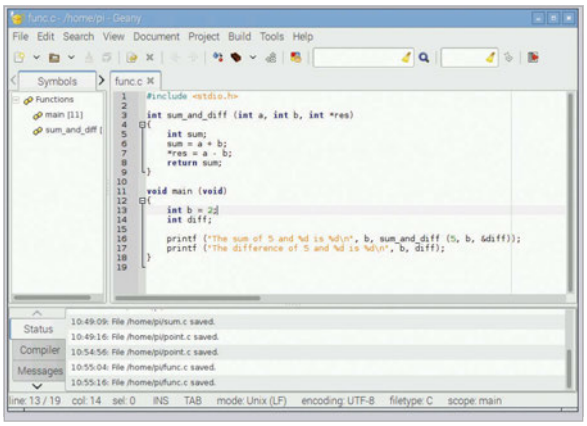
FUNCTIONS

Functions – how to split your code up into easy bite-size chunks...

All the examples we've looked at so far have had one function, **main**, with all the code placed within it. This is fine for small programs, but it's not really practical when code gets bigger, and it wastes space if you need to do the same thing more than once. Splitting code into functions makes it more readable and enables easy reuse.

RETURNING VALUES

A function can return a single value, or no value at all. If you define the function as returning void, there's no need to use a return statement in it, but you'll get an error if you don't include a return of the correct type in a non-void function.



Above You can use a function call wherever a variable of the same type as the value returned by the function could be used – in the code here, a call to `sum_and_diff` replaces an integer value in the arguments to `printf`

We've already seen functions used; the main function is a standard C function, albeit with a special name. We've also seen the **printf** function called by our examples. So how do we create and use a function of our own? Here's an example:

```
#include <stdio.h>
int sum (int a, int b)
{
  int res;
  res = a + b;
  return res;
}
void main (void)
{
  int y = 2;
  int z = sum (5, y);
  printf ("The sum of 5 and %d is %d\n", y, z);
}
```

This example includes both the **main** function and a second function, called **sum**. In both cases, the structure of the function is the same: a line defining the value returned by the function, the function name, and the function arguments, followed by a block of

code enclosed within curly brackets, which is what the function actually does.

What's in a function?

Let's look at the **sum** function:

```
int sum (int a, int b)
```

The definition of a function has three parts. The first part is the type of the value returned by the function: in this case, an **int**. The second part is the name of the function: in this case, **sum**. Finally, within round brackets are the arguments to the function, separated by commas, and each given with its type: in this case, two integer arguments, **a** and **b**.

The function's code is between the curly brackets:

```
int res;
```

This declares a local variable for the function, an integer called **res**. This is a variable which can only be used within the function itself. Variables declared within a function definition only exist within that function; if you try to access **res** from **main**, you'll get an error.

```
res = a + b;
```

Here, **a** and **b** are the arguments of the function. When a function is called, a local copy of the arguments is made. If you change the values of **a** or **b** within the function, that only affects the local values; it doesn't change the values that the arguments had in the function from which it was called.

```
return res;
```

Finally, we return the result. The function was defined to return an integer, therefore it must call the **return** statement with an integer to be returned to the calling function.

A function doesn't have to return a value; if the **return** type is set to **void**, it returns nothing. In this case, a function will return when it reaches the last line, but if you want to return early (in the event of an error, for example), you just call **return** with no value after it.

Calling a function

Let's look at how we call the function from **main**:

```
int z = sum (5, y);
```

The **sum** function returns an integer, so we set an integer variable equal to it. The arguments we supply to the function are inside round brackets, and in the same order as in the function definition: **a** is 5, and **b** is the value of **y**.

You can only return one value from a function, but you can also use pointers to pass multiple items of data back to the calling function. Take this example:

```
#include <stdio.h>
int sum_and_diff (int a, int b, int *res)
{
  int sum;
  sum = a + b;
  *res = a - b;
  return sum;
}
void main (void)
{
  int b = 2;
  int diff;
  printf ("The sum of 5 and %d is %d\n", b,
    sum_and_diff (5, b, &diff));
  printf ("The difference of 5 and %d is %d\n", b, diff);
}
```

The sum is returned as before, but we're also passing the difference back using a pointer. Remember that the arguments to a function are local variables; even if you change one in the function, it has no effect on the value passed by the calling function. By passing a pointer, the function doesn't change the value of the pointer itself, but it can change the value in the variable to which it points.

So we call the function with the same two arguments as before, but we add a third one, a pointer to the variable where we want to write the difference. In **main**, we call the function with the address of **diff** as the pointer argument to the **sum_and_diff** function; the difference is written into **diff** in the main function.

Order matters

One thing to bear in mind when defining functions is that the compiler reads files from top to bottom, and you need to tell it about a function before you can use it. In the examples above, this is automatic, as the definition of the **sum** and **sum_and_diff** functions is before the first call to them in **main**.

But in larger files, when multiple functions call multiple other functions, this gets complicated; it's not always easy to make sure the function definitions are all in the right order. To avoid this, C allows you to declare functions before they're used.

A function declaration is just the definition of the function, minus the function code within the curly brackets. So for the **sum_and_diff** function, the declaration would be:

```
int sum_and_diff (int a, int b, int *res);
```

Note the semicolon at the end! Function declarations are included at the top of the file. When the compiler finds a function declaration, it knows that at some point a function with this name, arguments, and return type will be defined; it then knows how to handle a call to it, even if it hasn't yet seen the definition itself.

VARIABLE SCOPE

If you declare a variable within a function, it's only usable within that function, not within any functions which call the function, or within functions called by the function. This is known as the scope of a variable: the parts of the code in which it is valid.

ARGUMENTS

A function can have any number of arguments, from zero up to hundreds. If you don't need any arguments, you list the arguments as `(void)` in the function definition (just like in the `main` function); when you call the function, just put a pair of empty round brackets `()` after the function name.

MIKE'S PI BAKERY

MIKE COOK

Veteran magazine author from the old days and writer of the Body Build series. Co-author of *Raspberry Pi for Dummies*, *Raspberry Pi Projects*, and *Raspberry Pi Projects for Dummies*.
magpi.cc/259aT3X

SLIDERS

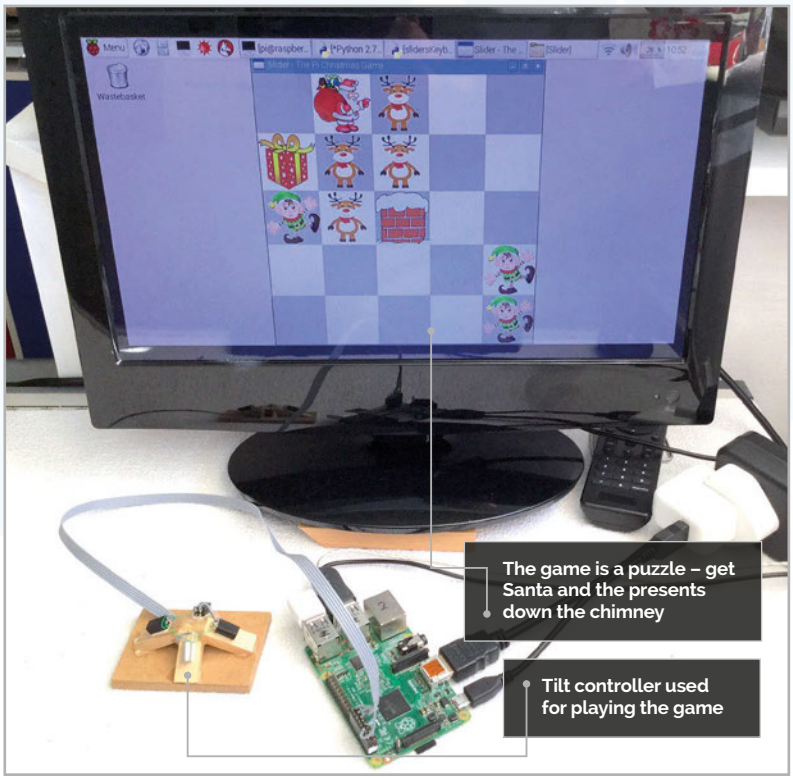
HELP SANTA AND THE CREW DELIVER THE PRESENTS

A simple but unique interface controls this Christmas puzzle

You'll Need

- 4× Tilt switches
- 5× ribbon cable
- MDF board
- Strip pine and glue

Santa has a problem, and he needs your help. He has landed on an ice-covered flat roof and the reindeers can hold firm, but Santa and the elves are sliding about all over the place. Can you help out by tilting the roof to guide Santa and his presents down the chimney? Make sure you don't lose an elf down it, though.



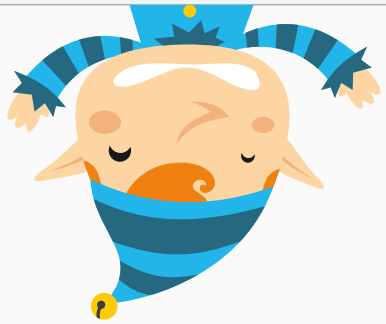
Slider game

The Sliders game is placed on a 5 by 5 grid; the object of the game is to deliver the presents down the chimney without losing any of your 'helpers'. Your cast consists of reindeer who helpfully (or not) stay fixed throughout the game, and elves who helpfully (or not) slide with Santa and his presents. Once you tilt the sensor, all the sliders slide in the same direction until they hit a reindeer, the end of the board, or another character. Once they're sliding there's no stopping them, no matter what you do with your tilt controller. As the chimney is at the centre of the grid, there's no way to get at it without the help of the other characters. The setup for each puzzle is graded into one of four difficulty levels, which range from easy to downright fiendish.

Physical interface

The controller is very simple. It consists of four tilt switches, arranged in a square base pyramid, so that normally none of the tilt switches are activated; that is, the ball is at the bottom of the tube. As this arrangement is tilted in a cardinal direction, just one of the tilt switches activates. Each switch is connected to a GPIO pin with the internal pull-up resistors enabled. The schematic is shown in **Fig 1**, and full details of the construction are in the steps overleaf.

You first need to get some images of each character, including the chimney, with a transparent background. These need to be 112 by 112 pixels in size. Clip art is a good source if your drawing skills aren't up to much. You can find the images we used



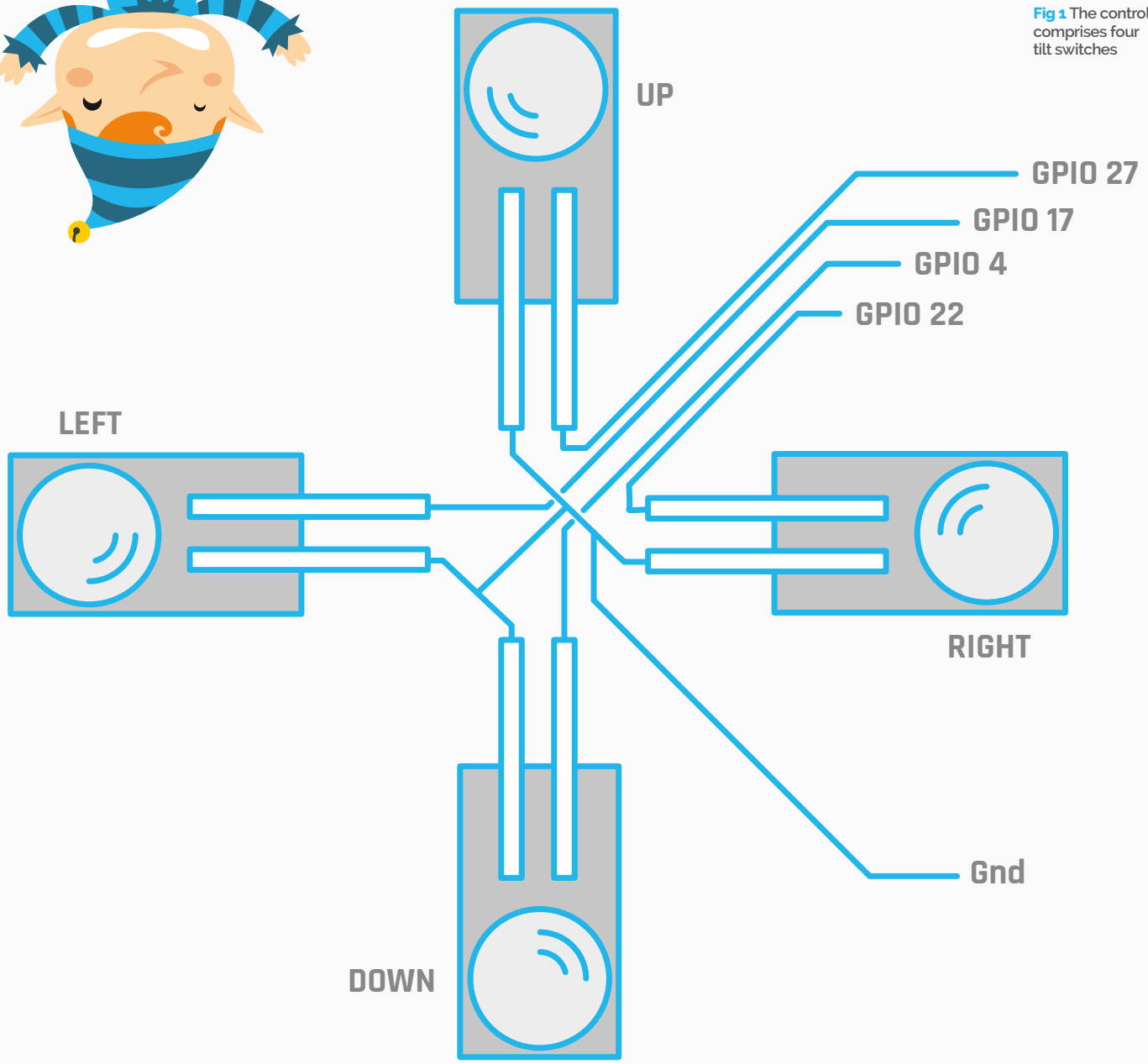


Fig 1 The controller comprises four tilt switches

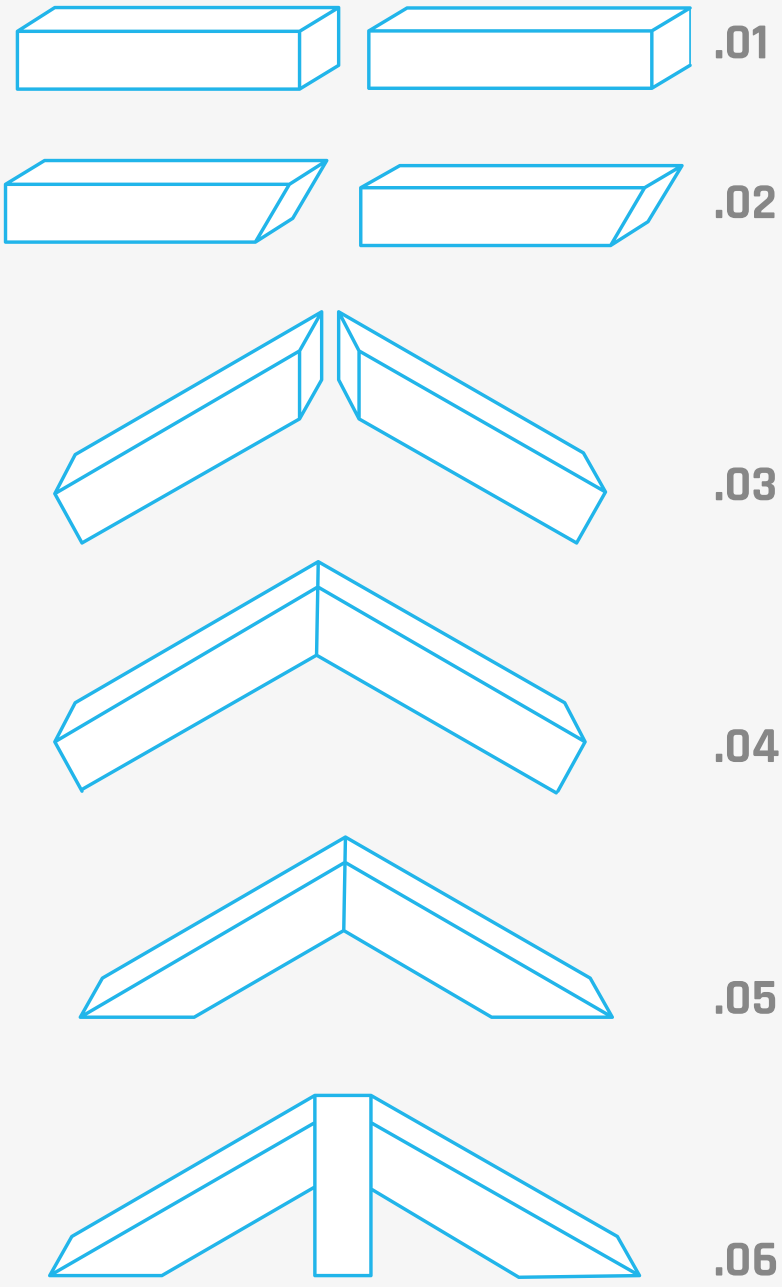
in our GitHub repository. You'll also need to get some sound samples: a short jingle of bells to accompany the sliding, and a triumphant finished sound. In addition, you'll need sounds for the presents and the elves plunging down the chimney, as well as a lost and won splash screen.

Create the software

The **Sliders.py** code listing is shown overleaf. It's written with the Pygame framework and the heart of it is the **`showBoard`** function. This draws a chessboard in a 5 by 5 grid, and places the character images based on the information in the board list. In fact, this was the first function we wrote, and we expanded the program around it. The board list is 25 elements long and shows what character number is in what square, with zero indicating no character.

The **`loadResources`** function assembles all the individual images into a piece list and, as the zeroth element is not used, we simply make it a copy of the first element, which saves complications further on in the code.

What makes the game look good is the smooth sliding to one end of the board. This can be a problem because sometimes a character can't move, and sometimes they can only move so far because other things are blocking them. The trick to fix this is to only move one adjacent square at a time. At the end of this, a list is made of all the characters that could move; if there are none then the slide is complete. Otherwise, the move list is applied again to one adjacent square. Thus, in a series of single moves, all the pieces end up in the right place, and the speed of the Pi makes it look like a smooth continuous



move. The move animation is done in a loop by redrawing the original and destination square of each moving character, and then drawing the character over this in a position slightly further over to the destination square. When the loop is completed, the character is in a new position, so the board list is then updated and the positions of the pieces checked to see if anything is in the chimney square, normally indicating the end of the game. A new move list is generated to see if any more sliding is needed. The only complication with this is that the adjacent square's position changes depending on the sliding direction, which means that the magic numbers have to be replaced by variables set when the slide direction is determined.

>STEP-01

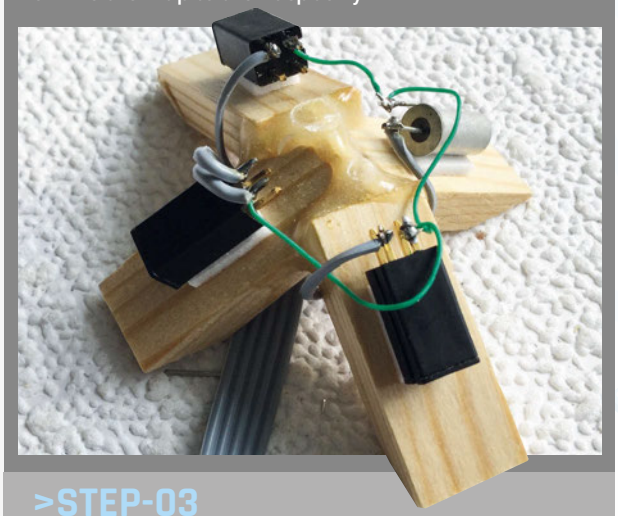
Make the switch support

Cut four pieces of 12×6mm strip pine 30mm long. Mitre the ends at a 60 degree angle; we used a disc sander for this. Take two pieces and glue them together at the mitre point. When set, make the other ends level by pushing it onto the disc sander. Take the other two pieces and glue them to the mitre joint of the first two, and again sand flat when set. Flood the whole joint with epoxy for added strength.

>STEP-02

Wiring up the switches

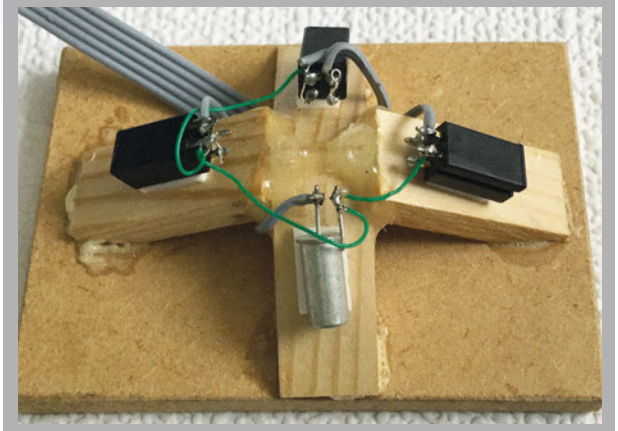
Wire up the switches to the ribbon cable. Put the cable on the back of the mount and bring up the wire to the top. Fix the ribbon cable on the underside with a double-sided foam pad. Use the same to fix the tilt switches to the arms of the switch support. At this stage test the switches, because wiring might be difficult after the next stage. You can do this with a continuity meter, or wire them up to the Raspberry Pi.

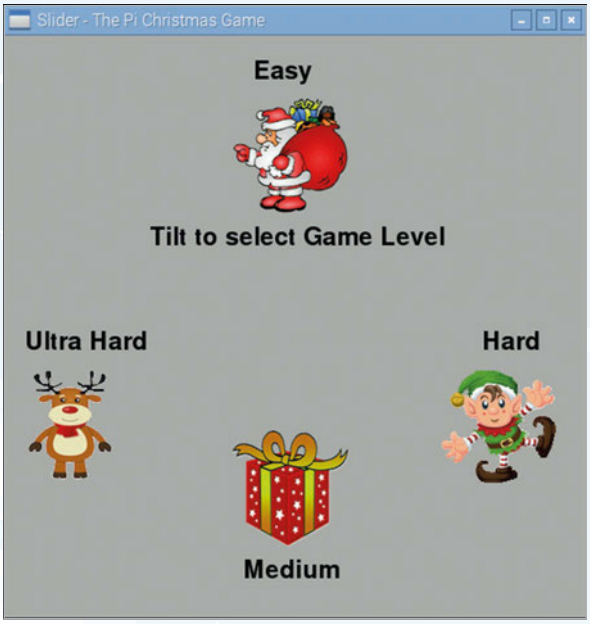


>STEP-03

Add the base

Cut a piece of 52×72mm MDF of 7mm thickness for the base and glue the switch support to it. Clamp them together while the glue sets. Make sure the switches are wired up to the correct GPIO pins, as shown in the photograph. Note that we used two different styles of tilt switch for the simple reason that this is what we had to hand. You might want to paint the finished unit.





Reading the tilt switches

The tilt switches are read individually, but are then placed into various bit positions of a single variable. This allows us to read the slide direction as a single number. With four switches, there are 16 combinations, but we're only interested in five: the four tilt directions and no tilt. This is done in the **getSensor** function, working with **getMove**.

The initial starting position for each game is defined in the **loadResorces** function and is split into four degrees of difficulty. You can choose what level of difficulty you want to play, and the setup position is chosen at random from that level. Pressing the **R** key will restore the original setup, while the space bar abandons a game and creates a new setup at the chosen level. Pressing the **RETURN** key will take you back to the level select screen. If you lose an elf down the chimney, you get to play the same game again.

Taking it further

The code printed here has 20 different setups; in the GitHub repository (**magpi.cc/1NqJjmV**) there's also a version with 40 different setups, as well as a version you can play using your keyboard to control the tilt with the cursor keys. You can customise the icons and the sounds very easily, or you can keep a track of the score. You might want to play each setup in order and not choose one at random, or, for the more advanced hardware people, even replace the tilt switches with an accelerometer.

Sliders.py

```
01. # Pi Christmas Sliders Game 20 setups
02. # By Mike Cook - October 2016
03.
04. import pygame, time, os, random
05. import wiringpi2 as io
06.
07. pygame.init()     # initialise graphics interface
08. pygame.mixer.quit()
09. pygame.mixer.init(frequency=22050, size=-16, channels=2,
    buffer=512)
10.
11. os.environ['SDL_VIDEO_WINDOW_POS'] = 'center'
12. pygame.display.set_caption("Slider - The Pi Christmas Game")
13. pygame.event.set_allowed(None)
14. pygame.event.set_allowed([pygame.KEYDOWN,pygame.QUIT])
15. screenSize = 560
16. screen = pygame.display.set_mode([screenSize,screenSize],0,32)
17. textHeight = 36 ; sq = 112 # square size
18. font = pygame.font.Font(None, textHeight)
19. random.seed()
20.
21. move = 0; moveInc = 0; moveIncX = 0; moveIncY = 0
22. tiltPins = [ 4,17,27,22]
23. squareC = [ (175,175,175),(255,255,255)]
24. board = [ 2,0,5,0,0, 3,0,0,0,0,  0,0,1,0,0,  0,0,4,0,0,
    0,0,2,0,3]
25. moveList = [0,0, 0,0, 0,0, 0,0, 0,0, 0,0]
26. gameOver = True ; newLevel = True ; sameGame = False
27. currentGame = 0 ; gameLevel = 0
28.
29. def main():
30.   global move, gameOver, newLevel
31.   loadResorces()
32.   initGPIO()
33.   print"The Pi Christmas Game"
34.   while True:
35.     checkForEvent()
36.     if gameOver:
37.       if newLevel:
38.         selectGameLevel()
39.         newLevel = False
40.       makeNewGame()
41.       waitLevel()
42.       #print "table level"
43.       time.sleep(1.0)
44.       gameOver = False
45.       lastMove = 0
46.       move = 0
47.     while (move == lastMove or move == 0) and (not gameOver):
48.       # wait for a move
49.       getMove(getSensor())
50.     lastMove = move
51.     if move!= 0 and not gameOver:
52.       gameSound[0].play()
53.     while move != 0 and not gameOver:
54.       # move all to the side of the tilt
55.       makeMoveList()
56.       s=0
57.       while s <= 1.0: # move free pieces one square
58.         showMove(s)
59.         s+= 0.05
60.       updateBoard()
61.       showBoard()
62.
63. def getSensor():
64.   global move
```

```
    sensor = 0
    checkForEvent()
    for pin in range(0,4):
      sensor = (sensor << 1) | (
io.digitalRead(tiltPins[pin]))
    return sensor

def selectGameLevel():
    global gameLevel
    drawLevelScreen()
    waitLevel()
    drawSelectScreen()
    getMove(getSensor())
    while move == 0:
      getMove(getSensor())
    gameLevel = move
    #print"game level ",gameLevel, " selected"

def waitLevel():  # hold until the platform is level
    table = getSensor()
    while table != 15 :
      firstLook = getSensor()
      time.sleep(1.0)
      secondLook = getSensor()
      if firstLook == secondLook :
        table = firstLook

def getMove(sensor):
    global move,moveInc,moveIncX,moveIncY,piece
    move = 0
    if sensor == 0xd: # up
      move = 1
      moveInc = -5
      moveIncX = 0
      moveIncY = -1
    if sensor == 0x7: # down
      move = 2
      moveInc = 5
      moveIncX = 0
      moveIncY = 1
    if sensor == 0xe: # right
      move = 3
      moveInc = 1
      moveIncX = 1
      moveIncY = 0
      piece[4] = santaFlip
    if sensor == 0xb: # left
      move = 4
      moveInc = -1
      moveIncX = -1
      moveIncY = 0
      piece[4] = santa

def drawLevelScreen():
    cp = screenSize/2
    pygame.draw.rect(screen,squareC[0],(0,0,screenSize,
screenSize),0)
    drawWords("Set platform level",cp-100,cp-100)
    pygame.display.update()

def drawSelectScreen():
    cp = screenSize/2
    pygame.draw.rect(screen,squareC[0],(0,0,screenSize,
screenSize),0)
    drawWords("Tilt to select Game Level",cp-140,cp-100)
    drawWords("Ultra Hard",20,cp)
    drawWords("Hard",screenSize-100,cp)
    drawWords("Easy",cp-40,20)
    drawWords("Medium",cp-50,screenSize- 60)
    screen.blit(piece[2],(10,cp+40))
    screen.blit(piece[3],(screenSize-140,cp+40))
    screen.blit(piece[4],(cp-60,60))
    screen.blit(piece[5],(cp-60,screenSize-180))
    pygame.display.update()

def makeNewGame():
    global board, gameOver, setup,sameGame, currentGame
    for s in range(0,25): # Wipe board
      board[s]=0
    if not sameGame :
      temp = currentGame
      while temp == currentGame:
        currentGame = random.randint(gameRange[gameLevel
-1]+1,gameRange[gameLevel])
    game = setup[currentGame]
    elements = len(game)
    #print"playing game ",currentGame
    for s in range(0,elements,2):
      board[game[s+1]] = game[s]
    board[12] = 1 # always the chimney
    gameOver = False
    sameGame = False
    showBoard()

def updateBoard():
    global board, gameOver, sameGame
    for c in range(0,12,2):
      if moveList[c+1] > 2: # not chimney or reindeer
        board[moveList[c]] = 0 # remove piece
        board[moveList[c]+ moveInc] = moveList[c+1]
    # new piece position
    if board[12] == 3:
      gameOver = True # lose
      gameSound[1].play() # elf scream
      sameGame = True
      screen.blit(loseImage,(0,60))
      pygame.display.update()
      time.sleep(3.0)
      board[12] = 1
    if board[12] == 4 or board[12] == 5 :
      gameSound[2].play() # present delivery sound
      board[12] = 1
      finished = True
      for s in range(0,25):
        if board[s] == 4 or board[s] == 5:
          # see if anything left to deliver
          finished = False
      if finished :
        gameOver = True # win
        screen.blit(winImage,(0,60))
        pygame.display.update()
        time.sleep(2.0)
        gameSound[3].play() # finish
        time.sleep(2.0)

def makeMoveList():
    # add to move list if not wrap round/space to move into
    global moveList,move
    c = 0
    for i in range(0,12):
      moveList[i] = 0 # remove all moves
    for square in range(0,25):
      if board[square] > 2:
        if not findWrap(square):
          if board[moveInc + square] < 2:
            moveList[c] = square # place piece
            moveList[c+1] = board[square] # piece type
            c += 2
    if c == 0:
      move = 0 # finished moving
```

Language
>PYTHON 2

DOWNLOAD:
magpi.cc/1NqJjmV

PROJECT VIDEOS

Check out Mike's Bakery videos at:
magpi.cc/1NqJnTz

```
def findWrap(square):
  wrap = False
  y = int(square / 5)
  x = (square - (y * 5))
  xw = x + moveIncX
  yw = y + moveIncY
  if xw>4 or xw <0 or yw > 4 or yw < 0:
    wrap = True
  return wrap

def showMove(movePart): # move all on the move list one square
  for c in range(0,12,2):
    if moveList[c+1] !=0:
      y = int(moveList[c] / 5)
      x = (moveList[c] - (y * 5))
      if moveList[c+1] > 2 :
        drawMove(moveList[c+1],x*sq+(sq*moveIncX*movePart),y*sq+(sq*moveIncY*movePart),moveList[c],x*sq,y*sq)
  pygame.display.update()

def drawMove(character,x,y,square,sx,sy):
  pygame.draw.rect(screen,squareC[square & 1],(sx,sy,sq,sq),0)
  # from square
  pygame.draw.rect(screen,squareC[(square & 1)^1],(sx+(sq*moveIncX),sy+(sq*moveIncY),sq,sq),0) # to square
  screen.blit(piece[character],(x,y))

def showBoard():
  c = 1
  for x in range(0,5):
    xp = x * 112
    for y in range(0,5):
      yp = y *112
      c = c ^ 1
      pygame.draw.rect(screen,squareC[c],(xp,yp,112,112),0)
  for square in range(0,25):
    y = int(square / 5)
    x = (square - (y * 5))
    if board[square] != 0 :
      screen.blit(piece[board[square]],(x*sq,y*sq))
  pygame.display.update()

def drawWords(words,x,y) :
  textSurface = pygame.Surface((14,textHeight))
  textRect = textSurface.get_rect()
  textRect.left = x
  textRect.top = y
  textSurface = font.render(words, True, (0,0,0), squareC[0])
screen.blit(textSurface, textRect)

def initGPIO():
  try :
    io.wiringPiSetupGpio()
  except :
    print"start IDLE with 'gksudo idle' from command line"
    os._exit(1)
  for pin in range (0,4):
    io.pinMode(tiltPins[pin],0) # mux pin to output
    io.pullUpDnControl(tiltPins[pin],2) # input enable pull up

def loadResorces():
  global gameSound, piece, winImage, loseImage, santa, santaFlip, setup, gameRange
  chimney = pygame.image.load("images/chim.png").convert_alpha()
  elf = pygame.image.load("images/elf.png").convert_alpha()
  rainD = pygame.image.load("images/rain1.png").convert_alpha()
  santa = pygame.image.load("images/santa.png").convert_alpha()
  santaFlip = pygame.image.load("images/santaFlip.png").convert_alpha()
  present = pygame.image.load("images/present.png").convert_alpha()
  piece = [chimney,chimney,rainD,elf,santa,present]
  winImage = pygame.image.load("images/win.png").convert_alpha()
  loseImage = pygame.image.load("images/lose.png").convert_alpha()
  soundEffects = ["bells","elfScream","HoHoHo","finish"]
  gameSound = [ pygame.mixer.Sound("sounds/"+soundEffects[sound]+".wav") for sound in range(0,4)]
  # Easy
  g1 = [4,0,2,1,2,22];
  g2 = [2,0,4,1,3,2];
  g3 = [3,0,4,1,2,2,3,20,5,21]
  g4 = [2,0,4,4,5,24,3,20]; g5 = [3,0,3,5,3,2,4,1,5,7,2,6]
  # Medium
  g6 = [3,0,2,1,4,5,5,6,2,7,2,9];
  g7 = [2,1,4,6,2,7,5,8,2,13,2,21]
  g8 = [2,0,2,1,2,2,2,6,2,7,4,20,5,21,3,24];
  g9 = [2,17,2,18,3,22,4,23,2,24]
  g10 = [2,0,3,1,3,4,4,9,2,14]
  # Hard
  g11 = [2,0,2,2,2,4,2,11,2,17,2,22,4,19,3,24]
  g12 = [2,2,4,3,3,4,2,6,5,9,3,24]
  g13 = [2,2,4,5,2,10,2,20,3,21,2,24]
  g14 = [2,0,2,7,4,8,5,9,2,13,2,14,3,18,3,19]
  g15 = [2,0,2,9,2,13,4,14,5,19,3,18,2,17,3,24,3,23]
  # Ultra hard
  g16 = [2,1,2,2,2,17,2,21,4,16,5,23,3,22,3,18]
  g17 = [2,0,2,1,2,4,4,5,5,6,3,10,2,22]
  g18 = [2,20,2,22,2,23,2,17,2,13,4,18,5,24,3,19]
  g19 = [2,1,2,3,2,11,2,18,2,23,3,17,3,22,4,16,5,21]
  g20 = [2,2,2,7,2,10,4,1,5,5,3,6,3,3,3,15]
  setup = [g1,g1,g2,g3,g4,g5,g6,g7,g8,g9,g10,g11,g12,g13,g14,g15,g16,g17,g18,g19,g20]
  gameRange = [0,5,10,15,20]

def terminate(): # close down the program
   print "Closing down please wait"
   pygame.mixer.quit()
   pygame.quit() # close Pygame
   os._exit(1)

def checkForEvent(): # see if we need to quit
   global move, gameOver,piece,newLevel,sameGame
   event = pygame.event.poll()
   if event.type == pygame.QUIT :
       terminate()
   if event.type == pygame.KEYDOWN :
     if event.key == pygame.K_ESCAPE :
       terminate()
     if event.key == pygame.K_r :
       gameOver = True # abandon game
       sameGame = True # don't set new level
       move = 0
     if event.key == pygame.K_RETURN :
       gameOver = True # abandon game
       newLevel = True # set new level
       move = 0
     if event.key == pygame.K_SPACE :
       gameOver = True # start game again
       move = 0

# Main program logic:
if __name__ == '__main__':
   main()
```

HENRY BUDDEN

Following the release of the Raspberry Pi when he was twelve, Henry has taught himself to code and use/break electronics.
magpi.cc/2eCbaMf

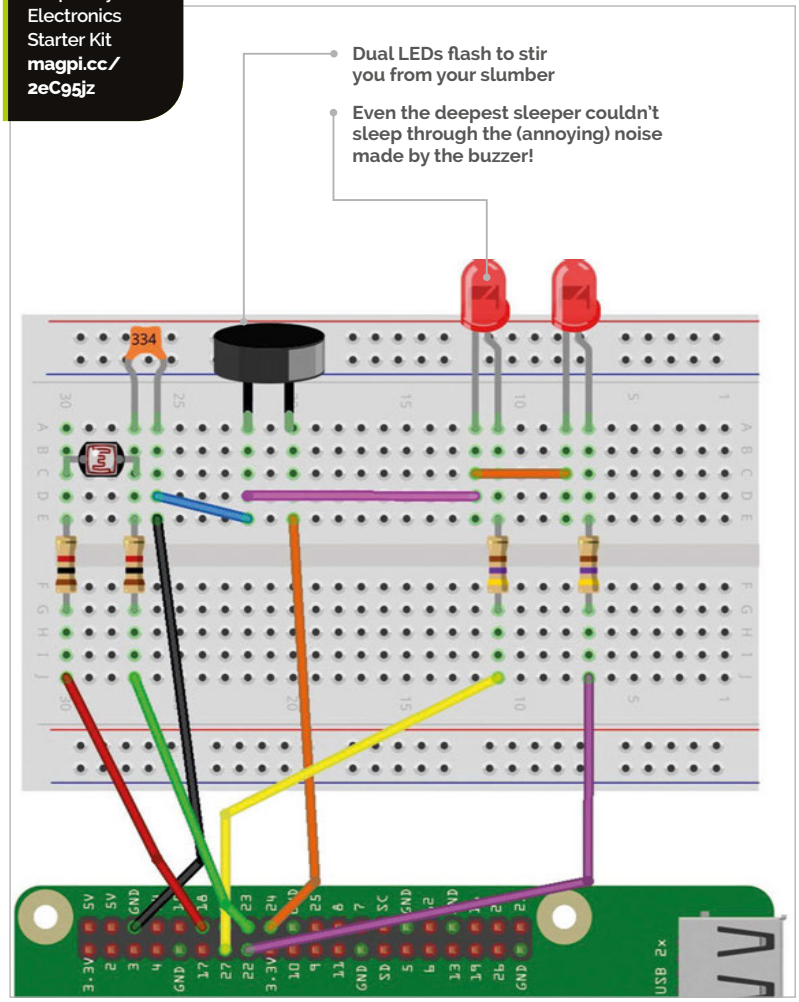
DIY SUNRISE ALARM

Use a breadboard and simple components to sense light levels and activate a loud (and blinky) alarm to wake you up each morning!

You'll Need

- 2× LEDs
- Light-dependant resistor (LDR)
- 330nF ceramic capacitor
- Buzzer
- 2× 1kΩ resistors
- 2× 470Ω resistors
- 6× Male-to-female jumper wires
- 3× Male-to-male jumper wires
- Or Monk Makes Raspberry Pi Electronics Starter Kit **magpi.cc/2eC95jz**

With the help of some exciting code and some clever electronics, you can use your Raspberry Pi to read an analogue signal from a photoresistor without the need for a conversion chip! Armed with this power, you can measure the ambient light level and trigger an effective alarm. The project is designed as an extension to the projects found in the Monk Makes Raspberry Pi Electronics Starter Kit.


Dual LEDs flash to stir you from your slumber
Even the deepest sleeper couldn't sleep through the (annoying) noise made by the buzzer!
334

>STEP-01

Connect the resistors

Once you have the components, begin to make the circuit by connecting up the resistors. Do this as shown in the circuit diagram, pushing each component's legs into the holes in the breadboard. Ensure that the bottom two resistors are 470Ω (yellow, purple, and brown), and the top two are 1kΩ (brown, black, and red).

>STEP-02

Add the rest of the components

Next, add the LEDs, making sure that the long legs are connected towards the bottom of the diagram, as shown. The flat side of the LEDs should be facing towards the 1kΩ resistors. When connecting the buzzer, the longest leg should be facing the bottom of the breadboard. Finally, connect the LDR and capacitor. These can be connected in any orientation.

>STEP-03

Connect to the Pi

Make sure your Raspberry Pi is turned off and unplugged before you do this. Using the three male-to-male jumper wires, connect the ground to the two LEDs, the buzzer, and the LDR as shown in the diagram. Next, use the remaining male-to-female jumper wires to connect the breadboard to the Raspberry Pi's GPIO pins.

>STEP-04

Install the software

Turn on your Raspberry Pi, and ensure it is connected to the internet. Using the Terminal, clone the GitHub repository containing the code to your Pi's SD card using the command:

```
git clone https://github.com/henrybudden/
rpesk-advanced/
```

Once the files have downloaded and you have returned to the command prompt, change into the

Language
>PYTHON 2

DOWNLOAD:
magpi.cc/2eCbCtZ

Sunrise_alarm.py

```
from Tkinter import *
import RPi.GPIO as GPIO
import time, math

GPIO.cleanup()

GPIO.setmode(GPIO.BCM)

sunrise = 50

a_pin = 18
b_pin = 23

buzzer_pin = 24
red_pin1 = 27
red_pin2 = 22

GPIO.setup(buzzer_pin, GPIO.OUT)
GPIO.setup(red_pin1, GPIO.OUT)
GPIO.setup(red_pin2, GPIO.OUT)

def discharge():
  GPIO.setup(a_pin, GPIO.IN)
  GPIO.setup(b_pin, GPIO.OUT)
  GPIO.output(b_pin, False)
  time.sleep(0.01)

def charge_time():
  GPIO.setup(b_pin, GPIO.IN)
  GPIO.setup(a_pin, GPIO.OUT)
  GPIO.output(a_pin, True)
  t1 = time.time()
  while not GPIO.input(b_pin):
    pass
  t2 = time.time()
  return (t2 - t1) * 1000000

def analog_read():
  discharge()
  return charge_time()

def read_resistance():
  n = 20
  total = 0;
  for i in range(1, n):
    total = total + analog_read()
  reading = total / float(n)
  resistance = reading * 6.05 - 939
  return resistance

def light_from_r(R):
  return math.log(1000000.0/R) * 10.0

while True:
  GPIO.output(red_pin1, False)
  GPIO.output(red_pin2, False)
  light = light_from_r(read_resistance())
  print light
  x = 0
  if light > sunrise:
    GPIO.output(red_pin1, True)
    GPIO.output(red_pin2, False)
    while True:
      x = x + 1
      GPIO.output(buzzer_pin, True)
      time.sleep(0.001)
      GPIO.output(buzzer_pin, False)
      time.sleep(0.001)
      if x == 250:
        x = 0
        break
    GPIO.output(red_pin1, False)
    GPIO.output(red_pin2, True)
    while True:
      x = x + 1
      GPIO.output(buzzer_pin, True)
      time.sleep(0.001)
      GPIO.output(buzzer_pin, False)
      time.sleep(0.001)
      if x == 250:
        x = 0
        break
```

directory containing the code by using the command `cd rpesk-advanced`.

>STEP-05

Run the code

The time has come to run the code and test out the alarm! After checking that the circuit you have built is an exact replica of the one shown in the circuit diagram, and that it is connected to the Pi, run the command `sudo python 01_sunrise_alarm.py`. You can test that the alarm works by shining a torch at the photoresistor from within a fairly light room. The LEDs should start to flash, and the buzzer should sound. If this happens, congratulations!

>STEP-06

Make it your own

The default threshold for the alarm's activation is when the photoresistor reaches a value of 50, which works well for testing as described previously. However, in order to use the alarm to accurately detect the sunrise in the morning, this value can be changed by entering the file editor with the command `nano 01_sunrise_alarm.py` and then changing the value found on line 13. We would recommend that you use 30 for fairly accurate detection of dawn. Save this change by pressing **CTRL+X**, followed by **Y**, then the **ENTER** key. This code can now be run again as in step 05.

K.G. ORPHANIDES

K.G. has been tinkering with computers since the mid-'80s and writing about them since the late '90s. They've mostly survived the experience.
twitter.com/KGOrphanides

You'll Need

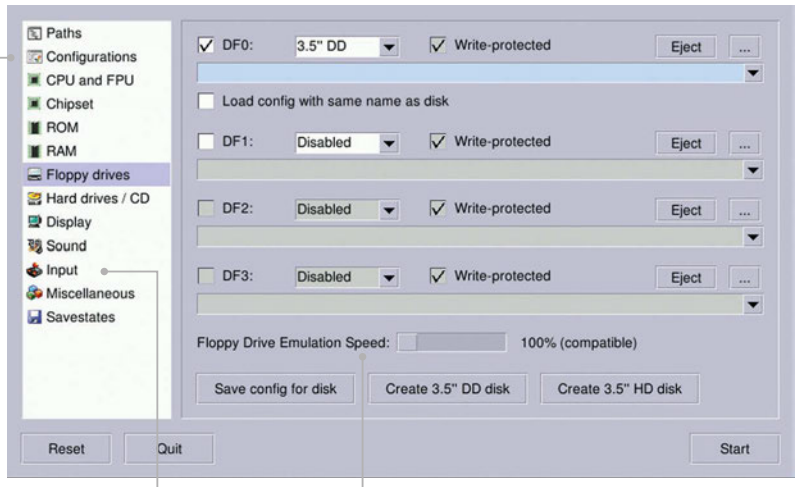
- microSD card
- USB stick
- Wired Xbox 360 controller
- Amiga Kickstart ROMs amigaforever.com
- Amibian bit.ly/Amibian

TURN YOUR PI INTO AN AMIGA

Recapture the glory days of 16-bit computing by turning your Pi into a faithful Amiga emulator

The Commodore Amiga's top-notch sound and graphics made it one of the most desirable home computers of the '80s and early '90s, at a time when your average IBM PC was still plodding along with EGA graphics and an internal beeper. Amiga games from the era have aged incredibly well, and look and play brilliantly on everything from a portable display to a widescreen TV. We'll take you through turning your Raspberry Pi 3 into a perfect modern-day Amiga emulator. You'll need a Windows, Mac OS X, or Linux desktop operating system to copy the Amibian Linux distribution to your SD card and unpack the Kickstart ROMs required to make it work smoothly.

Start by downloading the Amibian distro. Format a microSD card, decompress the Amibian RAR file, and use Win32DiskImager or Linux's **dd** command to copy the IMG file to the card. A 4GB card should be plenty, as Amibian only occupies around 300MB.



You can load and create emulated hardware configurations for specific Amiga computers

Game controllers, mouse, and keyboard configurations can be selected and tweaked in Input

Once you've set up your emulated hardware and firmware config, just mount a floppy disk image and click Start

Slot the microSD card into your Pi and power up. It'll boot directly into the UAE4ARM emulator, but there's some extra configuration to do before we start playing. Quit UAE4ARM to get to the command line and run:

```
raspi-config
```

Select Expand Filesystem, which will give you access to the entirety of the SD card's capacity for storage, then Exit and select Yes to reboot.

If your Pi won't output sound via HDMI properly, enter this at the command line:

```
nano /boot/config.txt
```

Make sure the following lines are present and aren't commented out with a preceding hash (#):

```
hdmi_drive=1
hdmi_force_hotplug=1
hdmi_force_edid_audio=1
```

Save and return to raspi-config:

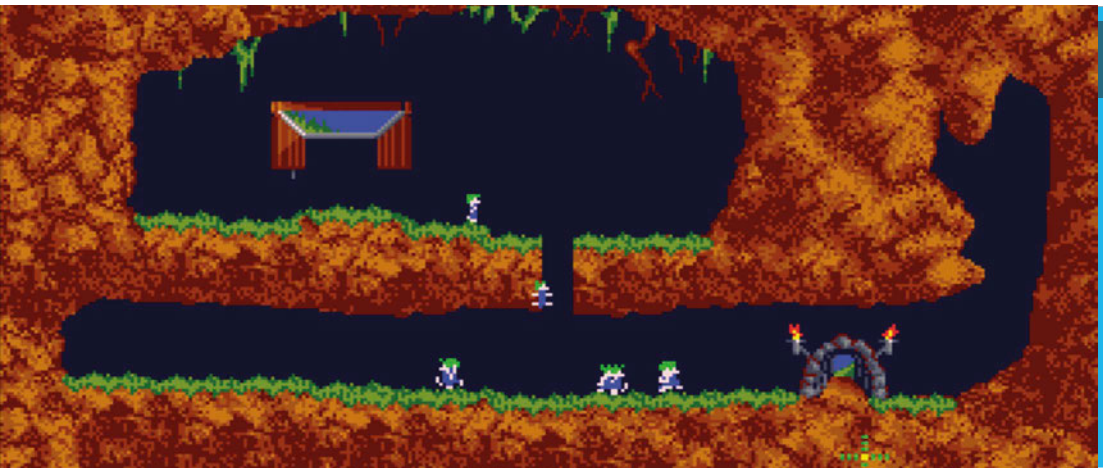
```
raspi-config
```

Select **Advanced Options > Audio > Force HDMI** and then reboot.

Kickstart me up

To run Amiga programs, you'll need a Kickstart ROM – firmware from the original computers. UAE4ARM comes with the open-source AROS ROM, which can run only some Amiga programs, so we recommend using genuine Amiga Kickstarts for reliability.

The Amiga's Kickstart ROMs and Workbench GUI are still being maintained, thanks to Italian firm Cloanto. Amiga Forever Plus Edition, priced at €29.95, gets you a complete, legal set of Kickstarts for every Amiga computer and console. Cloanto is still working



PUBLISHER-APPROVED GAME DOWNLOADS

Amigaland
amigaland.de

Ami Sector One
magpi.cc/2dDLEll

Dream17
dream17.info

Games Coffer
gamescoffer.co.uk

Gremlin Graphics World
magpi.cc/2dDKZ3S

on a Raspberry Pi edition, so you'll currently have to install Amiga Forever on a Windows PC or Wine and copy the files onto a USB stick.

There are other ways of obtaining Kickstart ROMs, but most are legal grey areas. You can extract them from an Amiga using a tool such as TransRom or find them on abandonware sites, but we strongly recommend supporting Cloanto's continued development of Amiga Forever.

Classic Amiga software is even easier to find. You'll get 50 games with Amiga Forever Plus, while some major publishers have made the Amiga versions of their games available for free (see above).

Many more games are only available online as legally dubious abandonware. They're easily found using any search engine, but inform yourself of the legal status of such software in your region before you download.

One true path

As Amibian doesn't include a window manager, it's easiest to download and copy everything to a USB stick using your operating system of choice. Helpfully, UAE4ARM can read Amiga ADF floppy images even if they're in a ZIP file.

We recommend copying everything to your microSD card. Fire up your Pi, exit UAE4ARM, and run:

```
mc
```

Copy your game files from **/media/usb** to **/root/amiga/floppys**, and your Kickstart ROMs, including a Cloanto rom.key file if you have one, to **/root/amiga/kickstarts**. Quit Midnight Commander and reboot:

```
reboot
```

In the latest version 1.313 of Amibian, two different versions of UAE4ARM are supplied. If you plan on using two Xbox 360 controllers, button mapping on controller two works best using the 'old' version, although the 'new' edition generally provides more options. To switch between the two, at the command line type either **rpiold** or **rpinew**. The following configuration instructions work with both versions.

Configure UAE4ARM

First, go to the Paths tab and click Rescan ROMs so UAE4ARM knows where to find everything.

The Configurations tab lets you select from several preset hardware emulations, with the default being an A1200 – just select and Load your chosen computer. You can tweak your virtual hardware in the CPU and FPU, Chipset, and RAM tabs.

Your configuration selection doesn't always set the relevant Kickstart ROM for you, so check the ROM tab, where you can choose Kickstarts from a pull-down menu. Note that many games require a specific ROM or hardware configuration to work properly, depending on which system they were originally released for.

To run most software, you'll need the Floppy drives tab. Just press the … icon next to drive DF0's Eject button, select the desired disk image, and click Start.

By default only drive DF0 is active, and most titles expect this configuration. To swap disks when prompted, press **F12**, eject the disk image in DF0, select the disk image you're asked for, and click Resume.

F12 will always pause and return you to UAE4ARM's main interface, so you can create a save state – a stored image of your progress in a game – or give up and load something new. The Reset, Quit, and Start/Resume buttons are always visible in UAE4ARM's GUI. Reset completely reboots your emulation and Resume returns you to your current game.

UAE4ARM automatically detects Xbox controllers. You can use two controllers for multiplayer gaming – if the second is unresponsive, you may need to press **F11** to disable your mouse and switch control to the pad. If you're running the 'new' version of the emulator, first select your controllers from the pull-down Port0 and Port1 menus in the Input settings.

Now you've got your Amiga emulator up and running, there's plenty of scope to build on the project, from setting up virtual hard disks to install Workbench and other software onto, to creating floppy images from your own original Amiga disks and using the Pi's GPIO to connect a classic '80s joystick.

JAMES HOBRO

James is a Pi hacker with current interests in robotic behaviour and multi-core programming. He has an alter ego as a research scientist at Schlumberger Gould Research in Cambridge.
github.com/jhobro

AN INTRODUCTION TO PARALLEL PROGRAMMING

Put all four cores in your Raspberry Pi to work by running tasks in parallel

We are used to the idea that the Raspberry Pi has revolutionised education and the maker community, but since the launch of the Pi 2 it has joined one of the biggest revolutions in the history of computing: the change from single-processor to multi-processor computers. This is fundamentally changing the way we think about writing programs. The Pi 2 and Pi 3 both contain quad-core processors, which means that they can run four tasks simultaneously. In principle, this gives us four times the original speed, but in practice it can be difficult to make use of this extra power. In this article, we introduce one of the simplest approaches to parallel programming that will enable you to make use of all the processing power on your Pi.

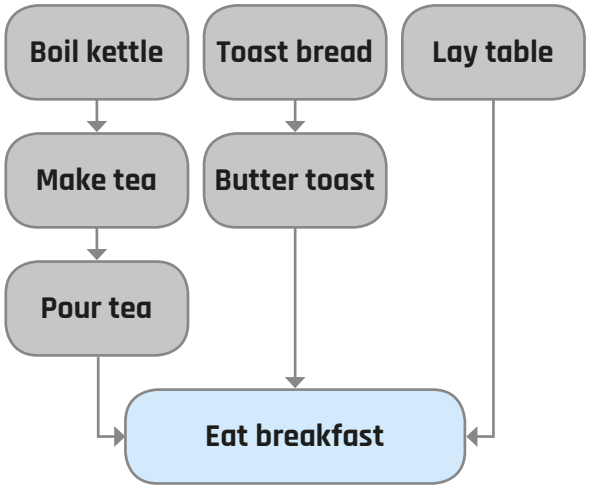
Three-core breakfast

Before we write any programs, let's look at something that most of us do every day: make breakfast. If we try to describe this in what might seem the obvious way – a list of instructions – it might look something like the list in **Fig 1**.

Fig 1
BREAKFAST LIST

Boil kettle
Make tea
Pour tea
Toast bread
Butter toast
Lay table
Eat breakfast

Fig 2 **BREAKFAST TASK DEPENDENCY GRAPH**



The breakfast list is a sequential program. It contains all the instructions necessary to make breakfast, but something important is missing. In reality, we would never perform these tasks in sequence – we would do some of them in parallel. After all, it's possible to boil the kettle and toast a slice of bread at the same time. But not all the tasks can be performed simultaneously – you can't make the tea until the kettle has boiled. How can we describe this? One way to do it is with a task dependency graph. Each task in the graph can only start once all the others it depends on have been completed. The breakfast task dependency graph (**Fig 2**) is a much more useful description than the sequential program; it's no coincidence that it describes more closely what we actually do when making breakfast, starting with all the tasks at the top of the graph and working our way down. As each of the initial tasks is completed, we can move on to the ones lower down that depend on it: once the kettle has boiled, we make the tea. Of course, not all programs that we'd like to run in parallel can be broken down in this way, but a task dependency graph offers a simple and powerful way to solve many problems in parallel.

Meet your maker

So how can we apply this to the Raspberry Pi 2 or 3? Raspbian contains a utility called Make, which follows a task dependency graph, executing tasks in the appropriate order and, where possible, running them in parallel using all four of the cores on your Pi. Make was designed to build programs in languages like C or C++ that require source code to be compiled to binary object files, executable programs, and libraries. However, it can be used to solve any problem that can be expressed as a set of tasks with dependencies. To use Make, we need to write a 'makefile' which describes how each task depends on the others. As an example, we will generate a collage of thumbnail images from a set of original image files. This is often required when dealing with large numbers of image files, for example when managing photos on

a server. As well as running Make, we're going to use ImageMagick for the image conversion, so ensure that both packages are installed on your Pi by running:

```
sudo apt-get update
sudo apt-get install make imagemagick
```

We're going to use Make first to create a thumbnail image for each of the originals, then finally combine them all in the collage. The majority of the work is in the generation of the thumbnails. The dependency graph (**Fig 3**) shows the relationship between each original image, its thumbnail, and the final collage. Since each thumbnail can be generated independently, we can put all four of our cores to work generating thumbnails, then finally create the collage once they're all ready.

Makefile rules

The makefile is all we need to describe our tasks and their dependencies. First, it defines the lists of original image files and thumbnails that we're going to work with. Then we define the two dependency rules in the graph. The first rule describes how to generate a thumbnail from a full-size image. It says that any file in the **fullsize** directory should be converted to a thumbnail with the same name in the **thumbs** directory, using the **convert** command. Note that the lines containing commands in a makefile must begin with a tab character (not spaces), shown in the listing as a long underscore. Our second rule specifies that the final collage depends on all the thumbnail images. It uses the **montage** command to create the collage and then displays it. Notice that we've told Make which commands will be used to rescale and combine the images, but we've not specified an execution order. Make will work this out for itself. We have moved away from writing a sequential program to describing a set of tasks and their dependencies, allowing the system to work out how to solve them most efficiently.

The simplest way to run our program is to type:

```
make
```

This will launch Make, which will read the rules we have specified in the makefile and get to work launching the tasks required in the correct order. However, by default Make assumes that only one core is available, so it will run each task sequentially. Luckily, we can easily tell it to use multiple cores:

```
make -j4
```

This command specifies that Make should execute up to four jobs simultaneously, and your thumbnails and collage will be generated up to four times faster than on a single core. Now try adding some new pictures to the **fullsize** directory or updating some of the existing pictures and rerun Make. It will now do only the work necessary to update your collage. Because Make understands the dependency graph and can see the date stamps on all the files in the graph, it can work out exactly which tasks are required and will not redo any work unnecessarily. This also means that you can stop it in the middle of a job and it will pick up the unfinished tasks correctly the next time you run it. Typing:

```
make clean
```

…will clean up by deleting the generated thumbnails and collage.

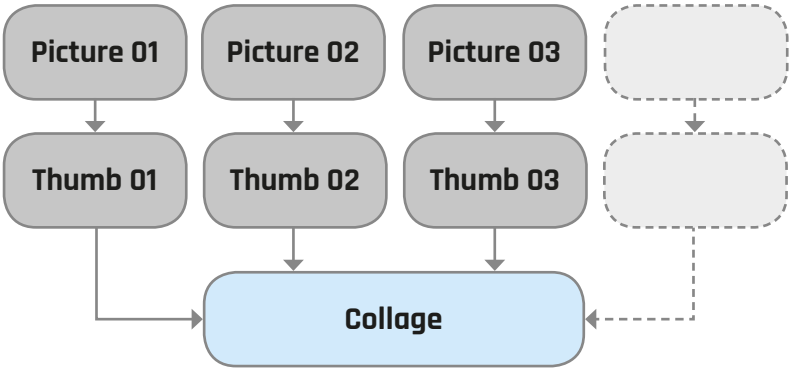
Makefile

Language >BASH

DOWNLOAD: magpi.cc/2bSdSJG

```
# Thumbnail size in pixels
SIZE = 128x128

# The list of original photos to use (fullsize/* refers to all files
# in the directory fullsize)
ORIGINALS = $(wildcard fullsize/*)

# Use the list of originals to build a list of thumbnails (this takes
# the list of originals and changes the prefix on each file from
# 'fullsize' to 'thumbs')
THUMBS = $(ORIGINALS:fullsize/%=thumbs/%)

# RULE 1: Generate each thumbnail from its original using the convert
# utility from ImageMagick, rotating the image if necessary
thumbs/% : fullsize/%
____convert $< -thumbnail $(SIZE) -auto-orient $@

# RULE 2: Combine all the thumbnails into the montage and display it
montage.jpeg: $(THUMBS)
____montage $(THUMBS) montage.jpeg
____display montage.jpeg &

# Clean up all thumbnails and delete the montage
clean:
____$(RM) thumbs/* montage.jpeg
```

Fig 3 COLLAGE DEPENDENCY GRAPH



WINFRIED PLAPPERT

Former IT consultant. Has enjoyed playing with Unix/Linux for many years. Received a Raspberry Pi 1 as a Christmas present a few years back.

USB hard disk attached to Raspberry Pi 2 with multiple operating systems

Use this like any normal multi-booting PC setup

MULTI-BOOT YOUR RASPBERRY PI 2

PART 01

You'll Need

- Das U-Boot
- Cross compiler tools

Use 'Das U-Boot' to boot multiple Raspberry Pi operating systems from a USB hard drive

Have your ever run different operating systems on your Raspberry Pi? Did you ever wonder about a bootloader for the Raspberry Pi? Did you ever think of having your root partition(s) on a USB hard drive? Here is a guide for all these ideas.

Over time, you might have collected a few Raspberry Pi 2 operating systems on various micro SD cards. Whenever you want to change to a different operating system, you might have to find the correct micro SD card, take out the current one, and put the new one in. This can be a fiddly job.

Using a USB hard drive and multiple operating system images, we can create a multi-boot device, much like on a normal PC.

For this we're going to be using Das U-Boot, which you can read up on here: **magpi.cc/2gaabm6**. This method will also work on other Pi models, but some of the FDT and kernel files will be different.

Get U-Boot working

All the compilation work has been done on Ubuntu MATE 16.04, but any Linux system should be fine for the job. You could install the source of Das U-Boot on the Pi itself and do the compiling there, but it would take a while.

First, download Das U-Boot in the terminal with:

```
git clone git://git.denx.de/u-boot.git
--depth=1
```

Next, install the cross compiler and tools:

```
sudo apt-get install gcc-arm-linux-gnueabihf bc
```

You'll need to set the essential environment variables on the PC:

```
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabihf-
```

Move to the U-Boot directory (**u-boot**) with **cd** and compile with:

```
make rpi_2_defconfig
make all
```

From the result we need exactly one file, called **u-boot.bin**, in the main directory. Copy this file to the boot partition of your micro SD card (**/dev/mmcblk0p1**) and modify **config.txt** in your boot

partition so it says **kernel=u-boot.bin**, or add the line if it's not there.

You're now ready to test your changes on the Pi 2. The webpages indicate that you need a serial console, but some modern U-Boots happily handle USB keyboards. Follow the instructions on your HDMI screen or your serial console. Interrupt the boot process of Das U-Boot when you're told to do so. You have two seconds to react, so be quick!

Type in this sequence of U-Boot commands:

```
setenv machid 0x00000c42
fatload mmc 0:1 ${fdt_addr_r} bcm2709-rpi-2-b.dtb
fatload mmc 0:1 ${kernel_addr_r} kernel7.img
```

Set the boot args:

```
setenv bootargs "bcm2708_fb.fbwidth=1920
bcm2708_fb.fbheight=1080 bcm2708_fb.fbdepth=32
bcm2708_fb.fbswap=1 dwc_otg.lpm_enable=0
earlyprintk console=tty1 console=ttyAMA0,115200
rootfstype=ext4 elevator=deadline rootwait
rootdelay=5 noinitrd root=/dev/ mmcblk0p2
smsc95xx.macaddr=AA:BB:CC:DD:EE:FF"
```

For manual booting, use the following:

```
setenv bootargs rootwait noinitrd rootdelay= 5
root=/dev/mmcblk0p2
bootz ${kernel_addr_r} - ${fdt_addr_r}
saveenv
```

Here's what the relevant terms mean:

machid
is the U-Boot ID for the Raspberry Pi 2 fatload.

bcm2709-rpi-2-b.dtb
loads a flattened Device Tree file into memory.

fatload … kernel7.img
loads the kernel into memory.

"bootargs"
is the U-Boot variable which contains the parameter string that's handed over to the kernel; you find this under normal circumstances in **cmdline.txt**.

Please note that we have to specify the **fbwidth=1920** and **fbheight=1080** parameters, otherwise you'll see an HDMI display in a 800×480 configuration. The normal Raspberry Pi 2 boot process works this out automatically, but when using Das U-Boot, the information generated by **start.elf** gets lost. Currently there's no easy way to recover this information successfully via Das U-Boot.

boot_via_uboot.txt

Language
>ASCII TEXT

```
# boot script to generate boot.scr
# this boot script will take the root device from the second
partition of the micro SD card

# machine ID
setenv machid 0x00000c42

# load FDT
fatload mmc 0:1 ${fdt_addr_r} bcm2709-rpi-2-b.dtb

# load kernel
fatload mmc 0:1 ${kernel_addr_r} kernel7.img

# set the boot parameters
setenv bootargs bcm2708_fb.fbwidth=1920 bcm2708_fb.fbheight=1080
bcm2708_fb.fbdepth=32 bcm2708_fb.fbswap=1 dwc_otg.lpm_enable=0
earlyprintk console=tty1 console=ttyAMA0,115200 rootfstype=ext4
elevator=deadline rootwait rootdelay=5 noinitrd root=/dev/mmcblk0p2

# boot the kernel
bootz ${kernel_addr_r} - ${fdt_addr_r}
```

Boot up

The Pi 2 should now boot into your favourite distribution. If it doesn't, you can fix it by modifying **/dev/mmcblk0p1/config.txt** and commenting out the **kernel=** line with a leading **#**. When it's working, you might get a slightly odd screen resolution. Let's fix that.

After the Pi 2 boots via Das U-Boot, you can automate the process by creating a **boot_via_uboot.txt** script file. This contains most of the commands you entered earlier and the full 'bootargs' parameter list:

```
setenv machid 0x00000c42
fatload mmc 0:1 ${fdt_addr_r} bcm2709-rpi-2-b.dtb
fatload mmc 0:1 ${kernel_addr_r} kernel7.img
setenv bootargs "bcm2708_fb.fbwidth=1920 bcm2708_fb.fbheight=1080
bcm2708_fb.fbdepth=32 bcm2708_fb.fbswap=1 dwc_otg.lpm_enable=0
earlyprintk console=tty1 console=ttyAMA0,115200 rootfstype=ext4
elevator=deadline rootwait rootdelay=5 noinitrd root=/dev/mmcblk0p2"
bootz ${kernel_addr_r} - ${fdt_addr_r}
# generate with ./tools/mkimage -O linux -A arm -T script -n "boot
schript for booting from usb hard disk" -d boot_via_uboot.txt boot1.scr
```

This file gets converted by the tool **mkimage** into a U-Boot executable script file. The exact command is:

```
./tools/mkimage -O linux -A arm -T script -n "boot script for booting
from micro SD card via u-boot" -d boot_via_uboot.txt boot.scr .
```

Copy the file **boot.scr** to the first partition of your micro SD card and test again. The system should automatically boot from the micro SD card.

Next time, we'll add partition scanning and create a boot menu you can use for true multi-booting!

FREQUENTLY ASKED QUESTIONS

NEED A PROBLEM SOLVED?

Email **magpi@raspberrypi.org** or find us on **raspberrypi.org/forums** to feature in a future issue.

Your technical hardware and software problems solved...

RASPBERRY PI GPIO

WHAT IS THE GPIO FOR?

Physical computing

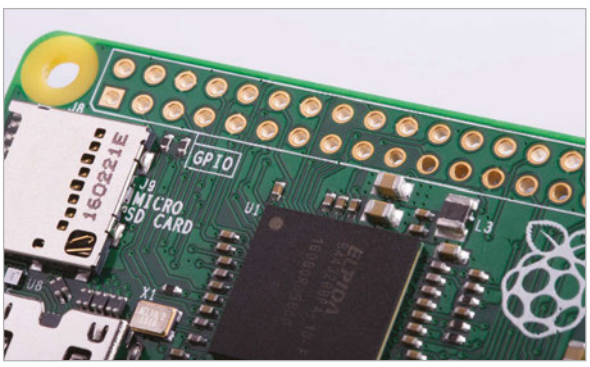
The GPIO pins are used to connect to the 'real world: components such as sensors, lights, and buzzers. These can then be programmed and controlled via the Raspberry Pi using built-in software.

Add-ons

The GPIO pins are also used to add functionality to the Raspberry Pi. Early on, LCD screens were a popular addition that connected via the GPIO. These days, HATs with many functions are available to add to the Pi.

Minor hacks

As well as physical computing, the GPIO pins act popularly as extra inputs. People add controllers to retro gaming projects through them. You can even power the Pi through the GPIO, allowing you to save some space on a bulky USB cable.



Right The Pi Zero features an unpopulated GPIO by default – you can add a header

WHAT DO THE INDIVIDUAL PINS DO?

Power pins

Throughout the 40 pins, there are a handful that provide power (5V or 3.3V) and a few more that can be use as a ground pin. Many of the individual pins can be programmed to output 3.3V.

GPIO

The vast majority of the pins are programmable in a number of ways. Most often you'll see them used to power simple components such as LEDs and buzzers, as well as sense when a button has been pressed.

Special functions

Some pins have extra functions, such as providing an SPI interface, access to I^2C, and more. These are used in more advanced projects, and you'll usually be told which pins to use and why they need to be used. For example, some chips need specific wiring to the GPIO.

HOW MANY PINS ARE THERE?

Original models

The original Raspberry Pis had 26 GPIO pins running down the side. This included all the power pins, some ground pins, and a number of programmable pins. You'll see a few add-ons for Pi still only using 26 pins so that they're backwards compatible.

B+ and beyond

Raspberry Pis now have a standard 40-pin arrangement. You'll find it on everything from the Pi Zero to the Pi 3. The first 26 pins use the same configuration as the original Pi so that old projects still work, but the 14 extra pins add more ground pins and programmable pins.

Compute Module

For some serious computer projects, the Compute Module is designed with real products in mind. The I/O board has 200 GPIO pins, as it's expected to do a little more than learning how to light LEDs with GPIO Zero!

FROM THE RASPBERRY PI FAQ
RASPBERRYPI.ORG/HELP

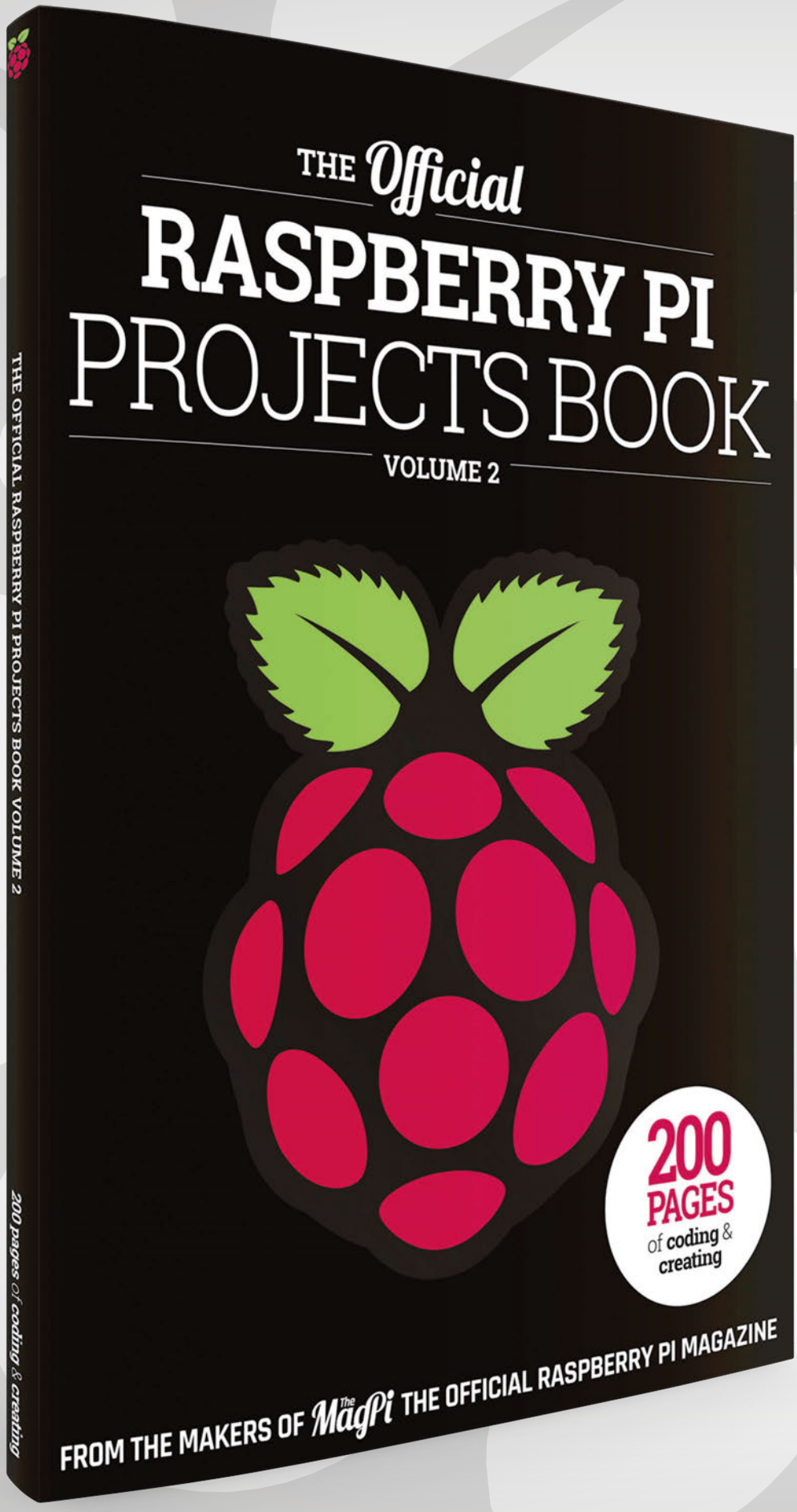
How powerful is the Raspberry Pi?
Real-world performance for models A, A+, B, and B+ is akin to a 300MHz Pentium 2, only with much better graphics – the GPU is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of general-purpose computing, and includes texture filtering and DMA infrastructure. The Model 2B is approximately equivalent to an Athlon Thunderbird running at 1.1GHz, but with higher-quality graphics thanks to the same GPU. The Model 3B is around 50% faster than the 2B.

Can it be overclocked?
Pi models A, A+, B, and B+ operate at 700MHz by default; most devices will run happily at 800MHz. The Model 2B operates at 900MHz by default and should run quite happily at 1000MHz. In the latest Raspbian distro, there's an option to change the overclocking options on first boot and at any time afterwards, without voiding your warranty, by running `sudo raspi-config`. It should be noted, however, that these are experimental settings and that not every board will be able to run stably at the highest setting. If you experience problems, try reducing the overclocking settings until stability is restored.

Do you need a heatsink?
You shouldn't need to use a heatsink, as the chip used in the Raspberry Pi is equivalent to that in a mobile phone, and should not become hot enough to need any special cooling. However, depending on the case you're using and the overclocking settings, you might find a heatsink to be advantageous. We recommend the use of a heatsink if overclocking the Model 3B.

What hardware interfaces does it have?
Depending on the model, the Raspberry Pi has either 40 or 26 dedicated GPIO pins. In all cases, these include a UART, an I²C bus, an SPI bus with two chip selects, I²S audio, 3V3, 5V, and ground. The maximum number of GPIOs can theoretically be indefinitely expanded by making use of the I²C or SPI bus.

WD PIDRIVE FOUNDATION EDITION FOR RASPBERRY PI
Offering USB functionality and an SD card preloaded with custom starter software, the WD PiDrive Foundation Edition is your ideal storage solution for Raspberry Pi. Start creating projects in a matter of moments.
WD PiDRIVE FOUNDATION EDITION
WDLABS
WDLABS
wdlabs.wd.com/holiday


THE Official
RASPBERRY PI
PROJECTS BOOK
VOLUME 2
200
PAGES
of coding &
creating
FROM THE MAKERS OF The MagPi THE OFFICIAL RASPBERRY PI MAGAZINE
THE OFFICIAL RASPBERRY PI PROJECTS BOOK VOLUME 2
200 pages of coding & creating

£12.99
200 pages of
Raspberry Pi

THE Official
RASPBERRY PI PROJECTS BOOK
VOLUME 2

Amazing hacking and making projects
from the creators of The MagPi magazine

Inside:

How to get started with Raspberry Pi

The most inspirational community projects

Essential tutorials, guides, and ideas

Expert reviews and buying advice

Available now **www.magpi.cc/MagPiStore**
plus all good newsagents and:

WHSmith BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

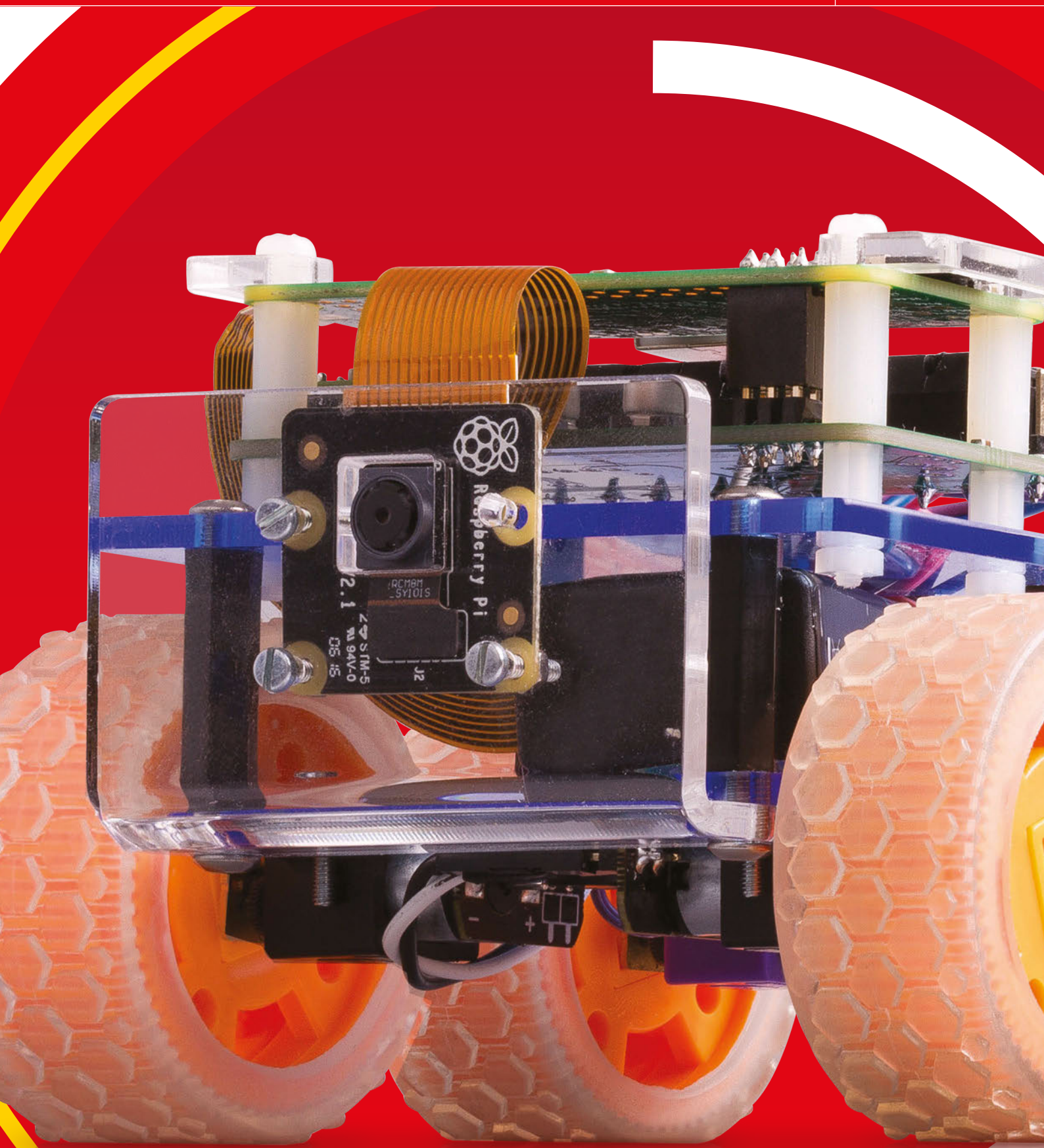
ADD SENSORS TO YOUR ROBOT

Take your robot from last issue and turn it into a true automaton

DOWNLOAD ALL THE CODE FOR THE ROBOT:

magpi.cc/2dx82hO

Now for the cool stuff: your Raspberry Pi robot can be a lot more than just a remote-controlled device. By using sensors, we can make your robot react to its environment. We also need to use sensors for the robot to be able to compete in Pi Wars autonomous challenges. In this feature we're covering the ever-popular line-following course, maze solving, and the straight-line robot drag race. The sensors we will be using are line-following and distance (range) sensors. We'll also take a look at alternative sensors you could add instead of the ones we've selected to use for the challenges.

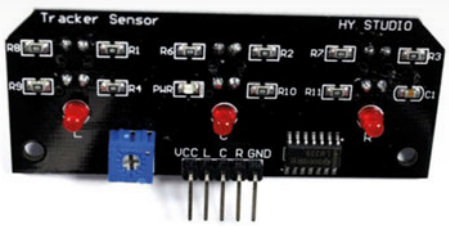

Raspberry Pi
2.1
J2

CHALLENGE: LINE FOLLOWER

This challenge is a favourite with many robot competitions. With great programming, amazing times can be achieved

The classic line-following challenge has been around for as long as people have been making robots and holding competitions. In this challenge, you must complete three laps around a circuit as fast as possible. The line will normally be black on a white background.

There are several methods used to sense the line. Infrared (IR) LEDs, the most common way to achieve this, use a light source and measure the light reflecting back. This works because the black line does not reflect as much light as the white background. The line-following sensor we have selected uses this method. The sensor has an array of three paired IR LED and photosensitive diodes; each pair of diodes is associated with an input pin on the Raspberry Pi.



LINE SENSOR

This Ryanteck line follower is the one we will be using. On the top side, you can see the three LEDs corresponding to the three sensors underneath the PCB. The adjustable resistor, commonly known as a pot, is for adjusting the sensitivity.

magpi.cc/2eoEXTl

THINGS TO CONSIDER

NOT ALL LINES ARE THE SAME!

The line in a line-following contest can be produced by several methods. This means it may be invisible to your sensor, as some inks can be transparent to IR light. Common methods are laser, ink, and electrical tape. Test your robot with different types of lines!

LET THERE BE LIGHT

The ambient light, or any shadows on the course, could affect how well your robot will run. It's best to test your robot in several different lighting conditions and with shadows falling across the course.

HIGHER, LOWER

The height at which the sensor is mounted can affect how well your sensor works. If you found your sensor is unreliable, it may be mounted too high or low. Try lowering the sensor if it's mounted high or raising if mounted low; repeat until you find the sweet spot.

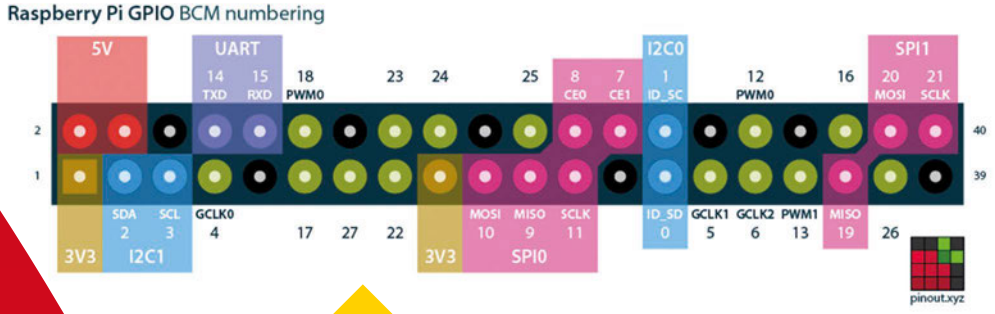
FASTER IS NOT BETTER!

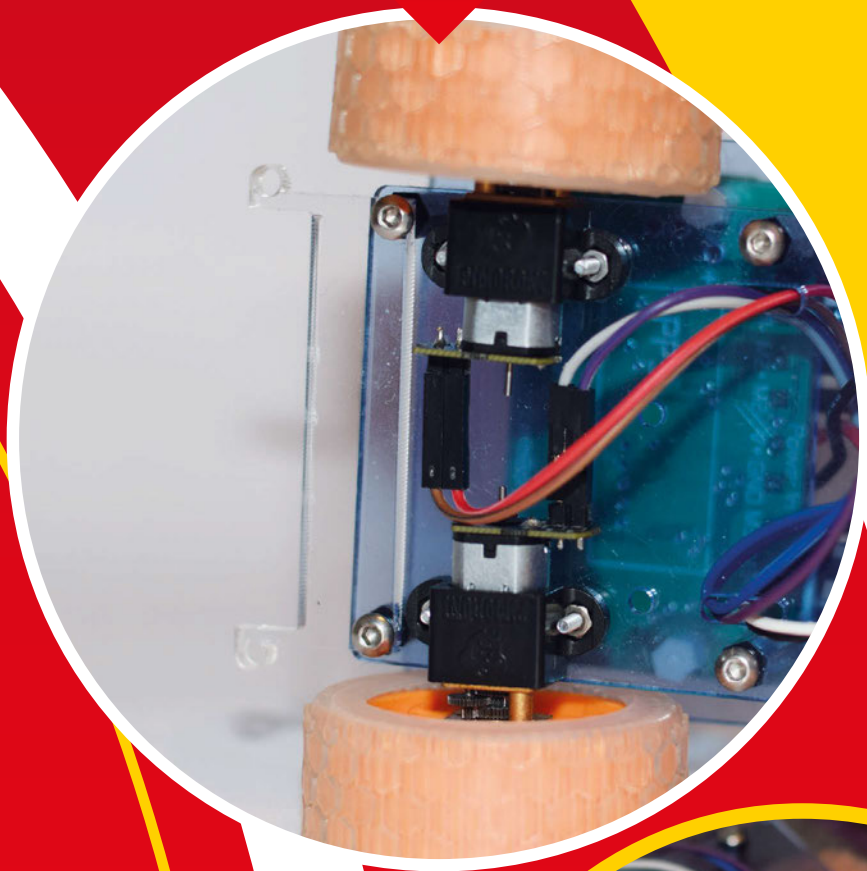
When you first try out your code, start with low speed settings for the motors, because if your robot is too fast, it will lose the line. Once you have proven your code, you can increase the speed of your robot until it fails, then go back to the last good settings.

>STEP-01

Mounting the sensor

The sensor needs to be mounted properly and allow you to adjust its height. We have designed a bracket to mount the sensor to our robot (find it here: **magpi.cc/2dx82hO**) and used PCB spacers to adjust the height of it. The sensor should be fitted to the front of the robot.



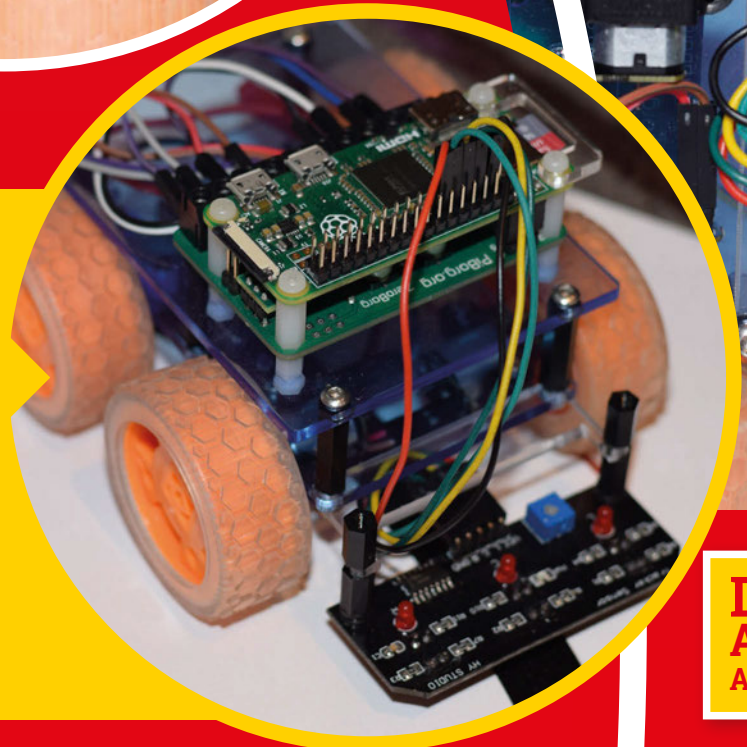


>STEP-02

Connect the sensor to the GPIO pins

For connecting the sensor to the Pi, you'll need five male-to-male jumpers. The robot should be powered down while connecting them. With the Pi Zero's USB ports towards you, count five pins from the left: this pin is GND (0V). The next three pins will be associated with the three output pins on the sensor, and the next pin is 3V3. Connect the jumpers to the corresponding pins on the sensor. See the table for assignments. Be careful connecting to the GPIO pins: double-check each one.

RASPBERRY PI	LINE-FOLLOWING SENSOR
GND	GND
BCM 17	L
BCM 27	C
BCM 22	R
3V3	VCC



>STEP-03

Testing the sensor

Now to test the sensor: power up your robot and place it on a white surface. The LEDs on the sensor will light. Now place it on a black line. Each time the following sensor goes over the line, the large LED corresponding to it should be unlit. If you've not done so already, download the code and test. If the line is on the left, the robot should turn to the right; if in the middle, the robot should move in a straight line; if on the right, it'll turn left.

DOWNLOAD ALL THE CODE AT: magpi.cc/2dx82hO

CHALLENGE: MAZE

The maze challenge is another classic and maze-solving robots are amazing – have a look on YouTube for some

This is a classic challenge, and very popular. The idea is to solve the maze in the quickest time possible; in the Pi Wars version, we are not allowed to touch the walls. For the challenge, we've selected a ToF distance sensor. The ToF (time-of-flight) sensor is an advanced version of the IR (infrared) distance sensor, which measures the intensity of the returning IR light. However, it's possible for the ambient light and the colour of the surface to affect readings from the sensors. ToF sensors were developed to counter those issues. They work by sending out pulsed light and count the time it takes for the light to return; this counters the effects of the colour of the surface and ambient light. Depending on how you approach this challenge, you may need to use more than one sensor.

ToF SENSOR

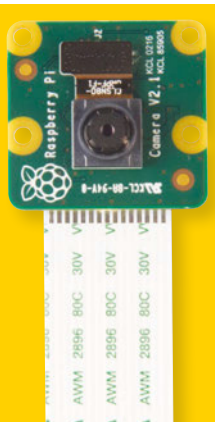
The time-of-flight sensor is the new kid on the block. The one we have selected uses I²C to talk to the Pi Zero and returns the distance in millimetres!

magpi.cc/2gclGsc

OTHER SENSORS

RASPBERRY PI CAMERA

The Raspberry Pi Camera Module can be used to navigate the maze, using computer vision techniques to find the walls or just working out which wall your robot is pointing at. The Python library you need for computer vision is OpenCV – or, if you find OpenCV overwhelming, there is SimpleCV. Think of SimpleCV as the CV version of GPIO Zero.



LIDAR

Lidar uses one or more lasers to map out the robot's surroundings; you can think of it as laser radar. The units cost many hundreds of pounds and are on the large size for our robot! A lidar may not be suitable for our robot, but we can dream.

ULTRASONIC

The ultrasonic sensor uses sound to find the distance to an object, the same way as a bat does. They work by measuring how long a pulse of sound takes to return to it. A benefit of using this sensor is that it's not affected by ambient light. However, it's affected by how dense the target object is, and the sensor has a wide target area which can lead to ghosting.

>STEP-01

Solder the I²C expansion header

Solder a right-angled header to the expansion port on the ZeroBorg. The expansion port is next to the DC-to-DC converter; the header should be on the top of the board and the pins need to face the outside of the board.

>STEP-02

Connecting the sensor

Be careful when connecting the sensor and make sure that your robot is powered down when doing so. The pins are not laid out the same as the other 2×3 header on the ZeroBorg; you only need to connect the 3V3, GND, SDA, and SCL. If you are planning to use more than one of the sensors, you will need to use a multiplexer like the Adafruit TCA9548A. For details on how to use the latter, see **magpi.cc/2dx82hO**, which has an example on how to read two sensors at the same time.

RASPBERRY PI OR ZEROBORG	POLOLU TOF SENSOR
3V3	Vin
GND	GND
SDA (BCM 2)	SDA
SCL (BCM 3)	SCL

maze.py

```
01  # pseudocode for reading distance sensor
02
03  listOfMoves = [ forward, right, forward, right, forward,
                    left, forward, left, forward]
04
05  function turn(move)
06    if move equals 'left' then set left motors forwards
                and right motors backwards until turned 90 degrees
07    if move equals 'right' then set left motors backwards
                and right motors forwards until turned 90 degrees
08    if move equals 'forward' then set left motors forwards
                and right motors forwards
09
10  for nextMove in listOfMoves
11    read sensor
12    if sensor less than 10cm then turn(nextMove)
13    sleep for 0.1 seconds
```

DOWNLOAD THE FULL CODE AT: magpi.cc/2dx82hO

HERE'S HOW IT WORKS...

CODE NOTES

Note: This is pseudocode for explanatory purposes – the full version is in the GitHub repo.

CODE LINE: 03

Defines a list of the required moves for the maze.

CODE LINES: 05 TO 08

Function for setting the motors for each type of move; for the turns, the robot will spin on the spot.

CODE LINES: 10 TO 13

The for loop reads each item of the move list and passes them to the function 'turn'.

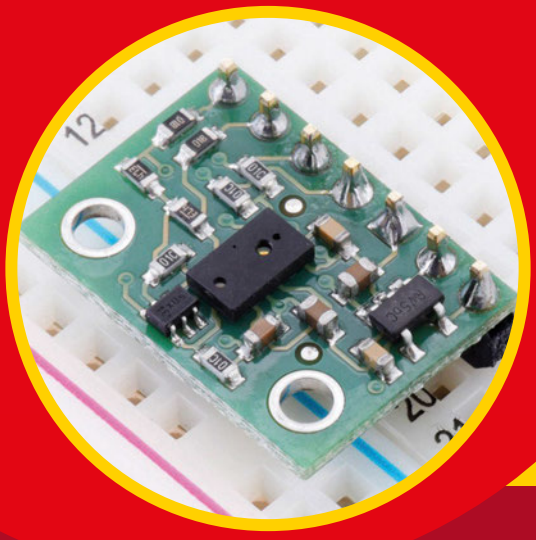
CHALLENGE: STRAIGHT-LINE SPEED TEST

The smell of burning rubber and screaming motors is autonomous robot drag racing. How fast can your robot cover seven metres in a straight line without hitting the wall?

The aim of the straight-line speed test – or, as we like to call it, robot drag racing – is to run the course as quickly as possible, and it's run autonomously. If you can afford it, it's best to use two sensors for both sides of the robot, and you may also want to change the motors to a lower ratio for more speed; 20:1 or 10:1 motors are available. The size of the wheels can also affect the speed and acceleration of your robot: small wheels are the quickest to accelerate while larger wheels have a higher top speed.

ToF SENSOR

We're using the same Pololu ToF sensor we used in the Maze challenge. The sensor is mounted at the front of the robot, pointing to the side, and needs to be below 40mm.
magpi.cc/2gclGsc



DIFFERENT SOLUTIONS

DISTANCE SENSOR METHOD

One or more distance sensors are mounted to the robot, facing the course wall(s). The robot is programmed to stay at a set distance from the wall(s) and if the robot gets too close to a wall, it's instructed to turn in the opposite direction. When only one sensor is used, the robot turns back toward the wall after a set time to ensure it doesn't hit the far wall.

MOTOR/WHEEL ENCODERS

This method counts the number of turns that a motor or wheel turns. The control program for this method allows each motor to run the same number of clicks from the encoders. When run for a short amount of time, this will help the robot run in a straight line.

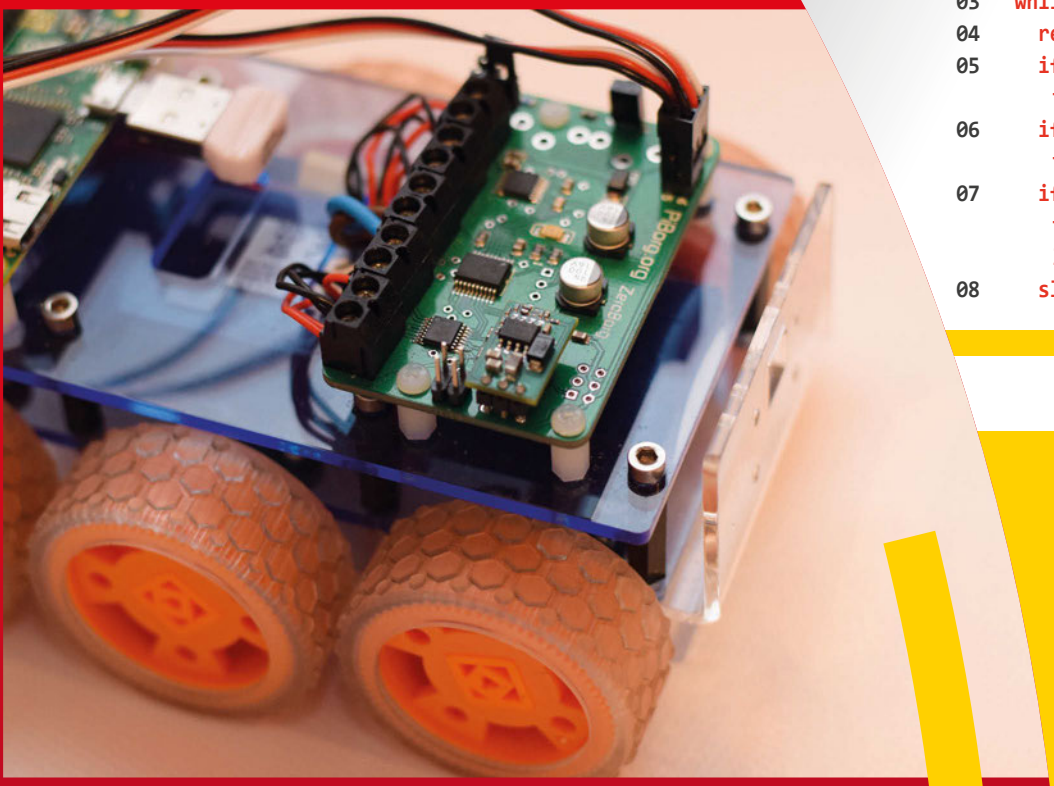
IMU (INERTIAL MEASUREMENT UNIT)

An IMU is a sensor that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers, gyroscopes, and magnetometers. This allows a bearing to be followed; you may want to add a means of aiming your robot before starting the challenge.

>STEP-01

Prepare the ZeroBorg

As before, you'll need to solder a right-angled header to the ZeroBorg. If you didn't do so in the last tutorial, you'll have to add it next to the DC-to-DC converter, and the header should be affixed to the top of the board with the pins facing away from the board.



>STEP-02

Wire up the sensor

The sensor is a delicate piece of equipment, so you need to be careful when putting it on. You'll need to turn the robot off before adding the sensor (although you should be doing this with any additions to the build). The layout of the pins is slightly different, so refer to the table below for what pins to wire up to the ZeroBorg. You won't need two as in the maze challenge, although if you want to add a second one for more precise readings, look at that spread for more information on using two ToF sensors.

RASPBERRY PI OR ZEROBORG	POLOLU TOF SENSOR
3V3	Vin
GND	GND
SDA (BCM 2)	SDA
SCL (BCM 3)	SCL

DOWNLOAD THE FULL CODE AT: magpi.cc/2dx82hO

HERE'S HOW IT WORKS...

straight-line.py

```
01  # pseudocode for straight-line speed test
02  start motors
03  while True
04    read sensor
05    if sensor returned value is less
       than 10cm then turn left
06    if sensor returned value is greater
       than 15cm then turn right
07    if sensor returned value is greater
       then 10cm and returned value is
       less than 15cm then drive forward
08    sleep for 0.1 seconds
```

CODE NOTES

Note: This is pseudocode for explanatory purposes – the full version is in the GitHub repo.

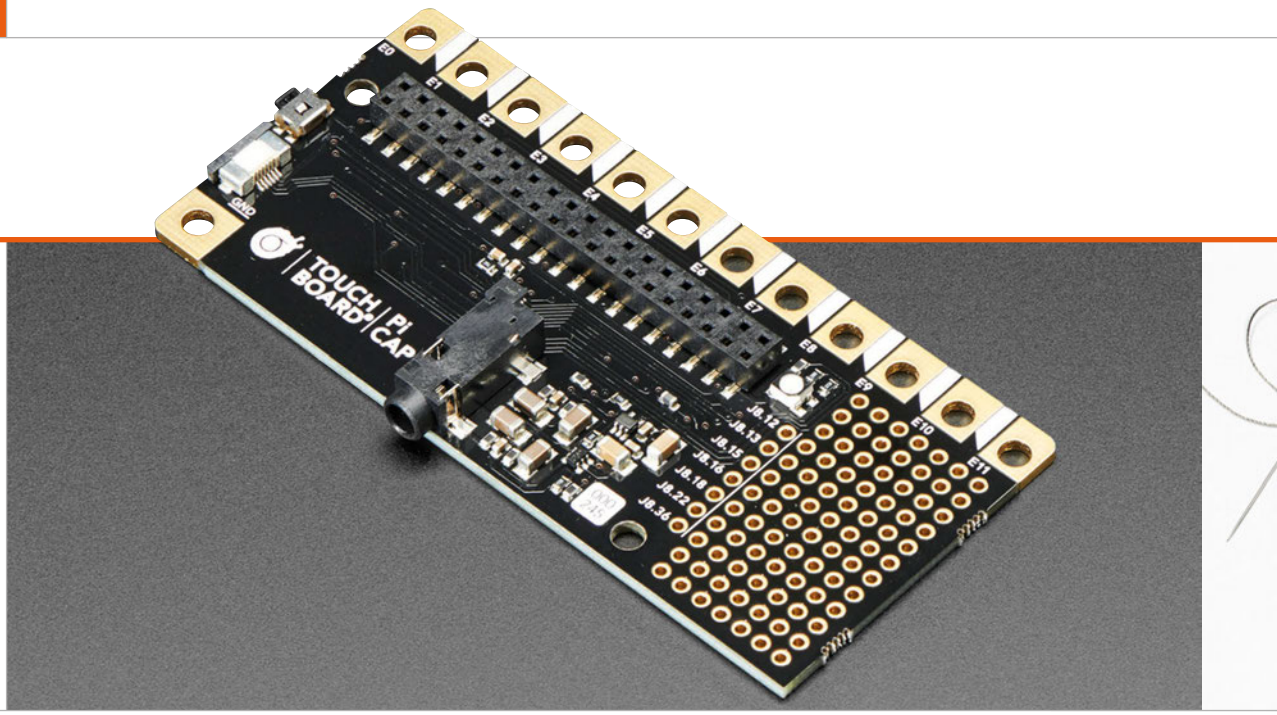
CODE LINE: 02
Starts the motors so the robot can move forward.

CODE LINES: 03 TO 08
The control loop. Runs as long the Raspberry Pi has power.

CODE LINE: 04
Reads the sensor.

CODE LINES: 05 TO 07
This sets the direction the robot will turn.

CODE LINE: 08
Adjusting this will affect how quickly the sensor is read.



Maker Says

"Connect your Raspberry Pi to the physical world

Bare Conductive

PI CAP

A HAT that adds some interesting features to a Raspberry Pi. Just how well does it fit?

Bare Conductive is one of those cool things we love to see stuff made with. The conductive paint can be used in some amazing creative builds and we love seeing people make things with it and post pictures and videos online.

To expand the uses of the paint, Bare Conductive the company has created its own special HAT for the Raspberry Pi, called the Pi Cap. HAT, Cap? The name is a bit more than just a pun, as one of the board's most interesting features is the addition of capacitive touch buttons/pads. We'll get to that, though – first, let's talk about the design.

The Pi Cap works very much like a standard HAT, sitting on top of your Raspberry Pi and granting you immediate access to more functions through the use of special software. Unlike most HATs, it hangs over the sides of the Raspberry Pi, though this is a deliberate design choice which allows easier access to some of its features. While it's designed with the Pi Zero in mind (the parts of the board that don't overhang fit snugly over the Zero's form factor), it will work on any other Pi model with a 40-pin GPIO.

The board comes pre-soldered so you can use it out of the box. You can put it straight onto a Raspberry Pi from there, although this does require some degree of software setup. The process is well documented on the website (**magpi.cc/2eKcB5C**), and you should be able to get it all set up within half an hour.

Internet of caps

With the Pi Cap on and ready to use, you have access to the aforementioned capacitive touch pads. These are the large gold connectors on the long edge, which can also be used to connect to wires and which are ideal for painting on with the Bare Conductive paint. Next to these is a large prototyping area with a GPIO breakout. There's also a physical button and an RGB LED attached to the board.

It's all really quite appealing. It adds a decent amount of useful functionality for education and for creating some interesting projects (you can find Capong in our last issue, but there are others on the firm's website under Suggested Tutorials: **magpi.cc/2eKcB5C**). We especially like the little breakout area, which is useful in any project. However, the capacitive pads are also excellent and the slightly larger holes in the connectors make them pretty good for wearables as well.

Related

RASPIO PRO HAT

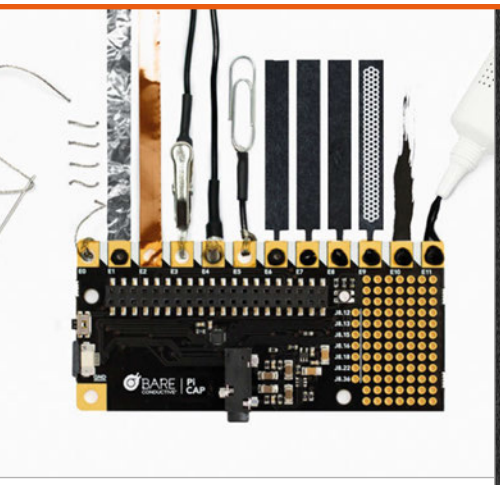
Although it doesn't quite have as many functions as the Pi Cap, it's still a great prototyping board for the Pi.

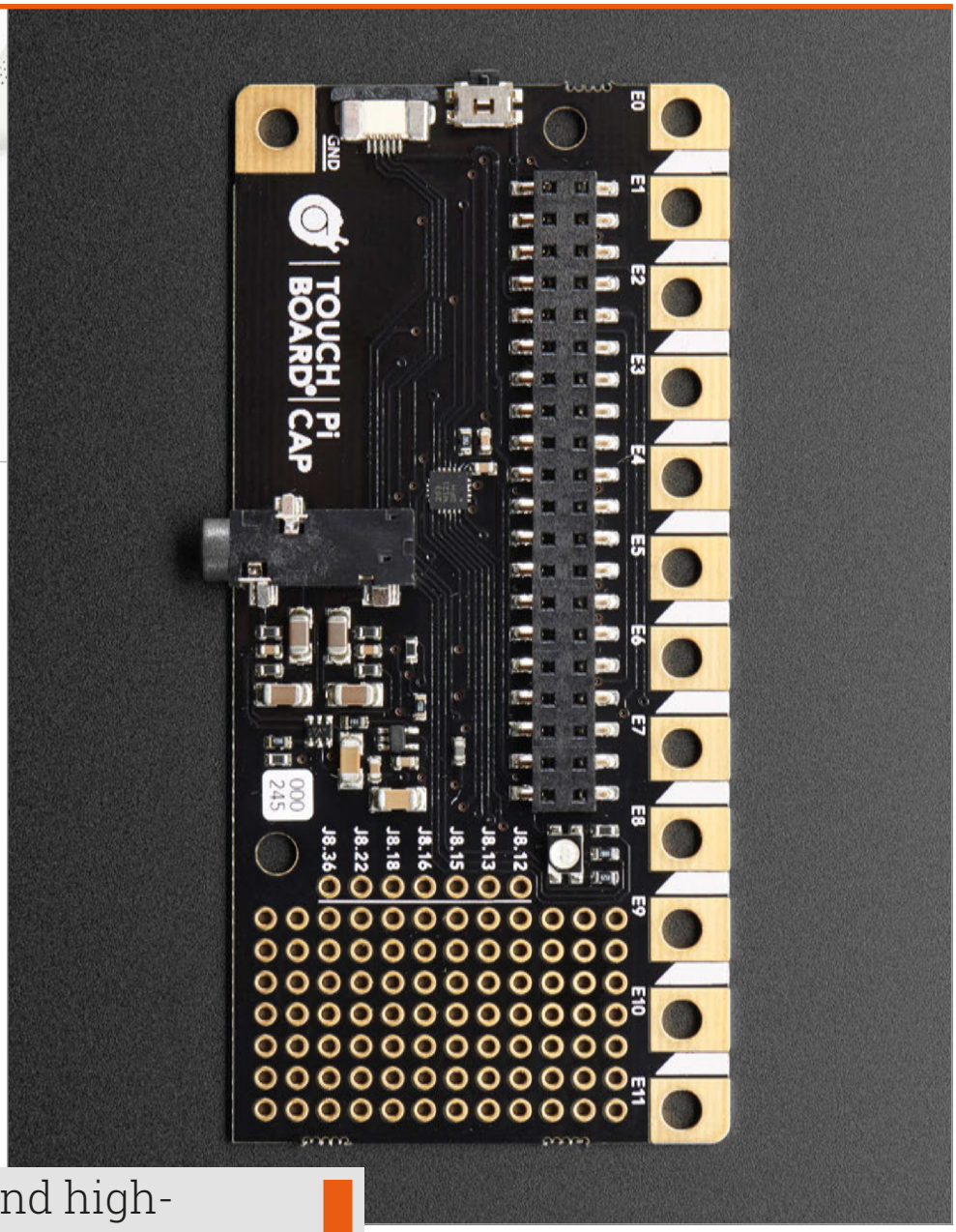
£15 / $19

rasp.io/prohat

magpi.cc/2dDDdXP

£28 / $40





Programming the Cap is pretty simple and you can do so in a number of languages, including the standard Python and C++, although you can also control it via Processing. Again, refer to our Capong tutorial from last issue for a sneak peek into how that works. There's also an interactive introduction and examples that come with the code so you can try some lower-barrier-to-entry tests; they're enough to get your head around what's going on at least.

Thinking cap

We do like the build of the Pi Cap. It's very robust and high-quality, possibly even more sturdy than the Pis themselves, which is quite the accomplishment. All the components are very small and have a very small form factor, meaning you will be unlikely to snap them off easily. There's even a high-quality audio jack installed on the board – it's perfect for the Pi Zero and is a little better than the 3.5mm jack on the Pi 3 as well.

> "It's very robust and high-quality, possibly even more sturdy than the Pis themselves"

We'd love to see an activity kit come with the Pi Cap in the future, with some components and perhaps a book to get you started with some fun projects. At the moment, though, it's still a pretty great board on its own, and maybe something to look into with Christmas coming up.

Last word

A neat little board with a lot of potential, adding some fairly unique features to the Pi. It would be better in a kit, but it's still great in its own right.

★★★★☆

Maker Says

"The Four Star package offers increased versatility"
Dremel

DREMEL 3000 FOUR STAR KIT

Can a toolbox and a selection of popular accessories renew interest in the Dremel 3000 rotary multitool?

Related

BLACK & DECKER WIZARD RT650KA KIT

While it lacks some of the more advanced accessories, like the SpeedClic system and flexible shaft, the Wizard kit is a considerably cheaper introduction.

£35 / $65
magpi.cc/2ghXtmn

Rotary multitools are in the family of devices that you never really think you need, then once you have one wonder how you survived without. At their heart, they're little more than a high-speed electric drill that you hold like a pencil; the wealth of accessories you can insert into the collet, though, enables them act like everything from routers and engravers to circular saws and polishers.

Dremel's latest attempt to turn people on to the joys of rotary multitools comes in the form of three new bundles, timed for pre-Christmas release. The Three Star Kit is a variant on the existing bundles, offering a Dremel 3000 multitool, a small number of accessories, and a plastic desk holster for storage. The Four Star Kit, as reviewed, ups the ante to include a wider range of accessories and a plastic toolbox for storage. The Five Star Kit, meanwhile, tops the range out with a fuller accessory selection and a more robust aluminium toolbox.

Unpacking the Four Star Kit reveals the first disappointment. The cardboard outer wrapping makes much of a bundled chess set, part of a competition run by Dremel encouraging users to make their own chess pieces using the included tool. The 'set,' though, is nothing more than a chess board printed on the reverse of the packaging and a set of black and white two-dimensional cardboard counters; this may be handy if you've forgotten what chess pieces look like, but it's not something you'd want to use to play a game.

The chess set is by-the-by, however. The kit's real meat comes inside the plastic toolbox, the lid of which entirely detaches with a flick of the solid-feeling blue catches. A second disappointment awaits, though: the interior of the toolbox is entirely plain, devoid of any indentation or padding to hold the tool and accessories properly. While it's certainly functional for storage, if you're planning to use your Dremel on the road you can expect a great deal of rattling and crashing of loose parts.

Thankfully, things improve from here on out. The kit includes many well-thought-out accessories: there's the flexible shaft, which provides a smaller tool head for easier use while insulating the user from the heavy vibration of the tool's motor at high speeds; there's the line and circle cutter, satistfyingly robust; and what Dremel bills as a 'multipurpose cutting kit' which includes three milling bits, a plastic guard, and a guide with pre-measured adjustable

dremel.com

£80 / $100

depth. All attach to the Dremel tool itself in moments and detach easily, though the flexible shaft does require replacement of the collet.

Accessorise

The remainder of the accessories, to make up the promised count of 55 on the outside of the box, run the gamut from cutting discs and sanding drums to polishing tools with a small vial of compound to match. Most can be installed quickly using the pre-installed EZ Twist Nose Cap, which replaces the traditional wrench for most tool-change jobs. The exception, as always, is in the cutting and grinding discs, most of which must be carefully screwed on to the bundled mandrel using the back part of the wrench tool.

Another notable accessory is a pair of discs designed around Dremel's relatively new EZ SpeedClic system. These will come as a revelation to anyone who has dropped a mandrel screw on the floor: each disc features a specially shaped and reinforced centre, which slots and locks onto a sprung mandrel. Discs can be swapped in seconds, rather than minutes, which is why it's such a shame Dremel only includes two SpeedClic discs in the pack; the remainder are of the traditional, screw-fixed variety.

In use, the Dremel 3000 – first released in 2011 as a replacement for the older Dremel 300, boasting improved cooling and better ergonomics – is a pleasure. The contours of the shape make it easy to hold, while the speed control – which adjusts from 5,000 to 32,000 revolutions per minute – is responsive and accessible. At high speeds the Dremel vibrates heavily and makes an awful din; fortunately, the majority of tasks are better carried out at lower speeds anyway, and for those that aren't, the Dremel can be hung from its metal loop and the flexible shaft can be used in its place.

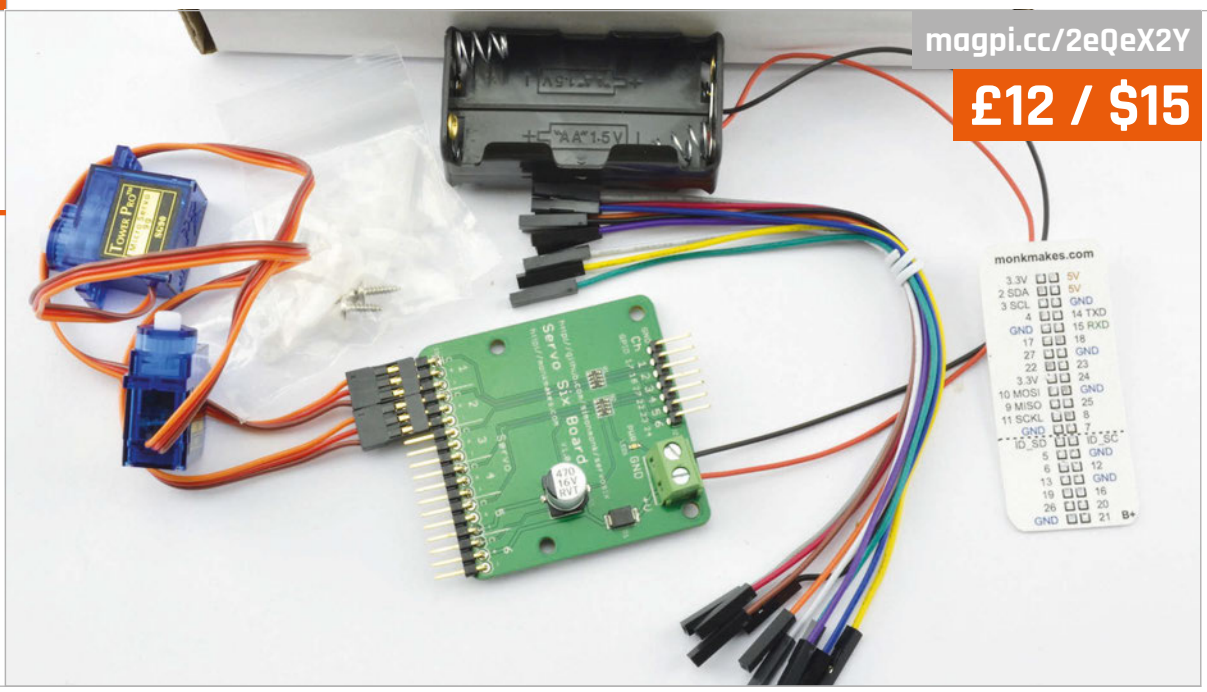
Last word

While the accessory choices may not be to everyone's taste, the Four Star Kit is a great introduction to the Dremel range at a reasonable price. A higher-quality toolbox would have been nice, though.

★★★★☆

magpi.cc/2eQeX2Y

£12 / $15



Maker Says

"Connect the two included servo motors to your Raspberry Pi"

Monk Makes

SERVO KIT

A kit for starting out with simple robotics projects on your Raspberry Pi

We see a lot of motor boards for the Raspberry Pi here at *The MagPi*, but very few servo controllers. On some level, they're very similar: they drive an electronic motor that can be programmed by the Pi. The major difference is the power and accuracy of the components; motors tend to just spin, while you can tell a servo to turn a certain amount. The limitations of both types can be worked around with a good bit of code and know-how, but it's best to use a dedicated driver for each.

The Servo Kit we're looking at here from Monk Makes comprises several parts, chief among them being the Servo Six board, a dedicated servo controller. The rest of the kit is a couple of simple servo motors, some jumper wires, and a battery case. It's enough to get you playing around with servos, with room for improvement as you learn.

It connects to the Raspberry Pi via seven pins (six GPIO and one GND). As it has so few connections, it doesn't function as a HAT; instead, you'll be connecting it via the jumper cables. This is better for a lot of projects, as it means the rest of the GPIO pins are available for other bits. You'll need to figure out a workaround if you do have a HAT as part of your project, though.

Construction is simple: there's no soldering involved, and there's plenty of info on the Servo Six page (**magpi.cc/2eXxxtr**) and GitHub repo for putting together a simple little circuit and programming it. The code uses Python, as usual, and instead of the simple forward, left, right, and stop commands from other Monk Makes motor boards, this one works in angles. Setting angles will cause the servo to spin to what it defines as that angle, but it will require calibrating for your project, or at the very least some trial and error. You'll have to make sure you understand which angle spins where, as well, when you want to move it 'back', and there are some little tricks you'll need to make use of to spin it in the correct direction sometimes.

The Servo Six Board itself is pretty great, though. You can control six separate servos with it and the code to differentiate between them is as simple as using the correct number, from 1 to 6, in the angle-changing code. With just the two servos included in the box, you'll need to look at getting more servos if you want to upgrade your projects, but it won't require any extra connections to the Raspberry Pi.

Last word

A great little kit for learning how to use servos that also has room to grow into some excellent projects. It's very well priced too.

★★★★☆

Related

ADAFRUIT MINI PAN-TILT KIT

A servo kit with more of a purpose, this can be modified to any applicable use.

£20 / $19

magpi.cc/2eQBU6a

magpi.cc/2fMEFHJ

£10 / $12

RASPBERRY SWEETS

We head to our favourite sweet shop, Mr Humbug's in Greenwich, and pick up a bagful of raspberry-flavoured sweets for Christmas

It's Christmas, and the team at MagPi Towers are benchmarking a lucky-dip mix of raspberry-flavoured sweets.

First up had to be Nerds. A hit of pure sugar, followed by a fresh minty afterglow with a mild taste.

It's often said that a camel is a horse designed by committee. We'd love to meet the committee which was responsible for naming Hubba Bubba Bubble Tape Sour Blue Raspberry. It looks like blue Sellotape and tastes of absolutely nothing. We had fun blowing blue bubbles with it, though.

According to MentalFloss (**magpi.cc/2fqQ41X**), sweet-makers of yore used a dark red dye called Amaranth E123 until the 1970s. But a Soviet study linked it to cancer and it was banned. They had lots of spare blue dye lying around and decided to use it for raspberry sweets.

Food for thought

AirHeads Chew has the scrummy tagline, 'Artificially Flavoured Candy'. We check out the nutritional facts: 1% fat, 0% sodium, 5% carbs, and 0% protein.

Inside, the AirHeads Chew is a brighter shade of neon blue than its metallic wrapper. It's tough and chewy. We suspect that it might be 94% blue dye. Bleugh!

To get rid of the taste, we crack open a can of Berry Fanta. Also blue. It contains 43g of sugar. In the interests of science, we measure out 43 grams of granulated sugar on some scales. It's a sobering sight. The soda tastes nice, though.

Our high hopes for Swizzels Sherbet Dip Tangy Raspberry didn't go unrewarded. The pink stick says 'Luv U' and 'Hug Me', a welcome change from the '90s 'Xtreme' styling of other sweets. We dip and dab the old-fashioned way. Pure sugar with a tangy edge.

In contrast, the Blue Razz Jawbreaker is the worst thing our reviewer has ever tasted. And she's drunk fermented horse milk in France and eaten grits from a shack in Alabama that had 'we got good eats' painted on the side. We're guessing kids gamble on who can mouth one the longest.

We saved the Barnetts Mega Sours for last, as Mr Humbug had warned us about them. They're sour, but not as bad as the Blue Razz Jawbreaker. We gamely give it a go for 30 seconds before getting heartburn and a headache. See you next year.

Last word

Sherbet Dip is a sweet treat that takes us back to a childhood where dabbing had a different meaning. Everything else was a bit too sour for our taste buds. The Blue Razz Jawbreaker was worth tasting just for the eye-watering experience.

★★★★★

Related

FRUIT

This seed-bearing structure contains fructose (naturally occurring sugar). Unlike table sugar, fructose also contains vitamins, minerals, and nutrients. The NHS advises that you should eat five pieces of fruit a day.

50p / $1

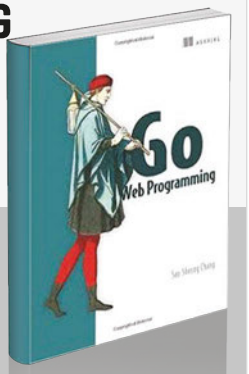
magpi.cc/2exjThS

RASPBERRY PI BESTSELLERS

A BETTER WEB

Don't settle for mediocre, bloated, and slow; find more effective ways to power the web!

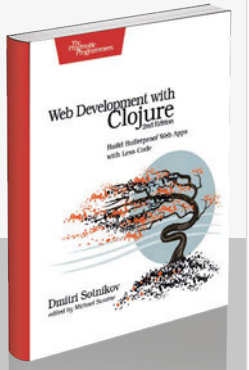
GO WEB PROGRAMMING



Author: Sau Sheong Chang
Publisher: Manning
Price: £27.99
ISBN: 978-1617292569
magpi.cc/2f0FF9a

From microservices to scalable, high performance web apps on multi-processor platforms, discover why Go's simplicity – its directness with HTTP – and standard library make for happy, productive programming.

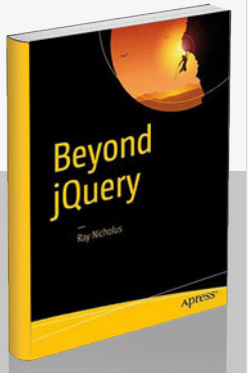
WEB DEVELOPMENT WITH CLOJURE



Authors: Dmitri Sotnikov
Publisher: Pragmatic Bookshelf
Price: £25.50
ISBN: 978-1680500820
magpi.cc/2f0E835

One of the best web dev books, updated. Build an app with the Ring/Compojure and Luminus framework, functional programming, and the flexible, replaceable components of the Clojure web stack.

BEYOND JQUERY

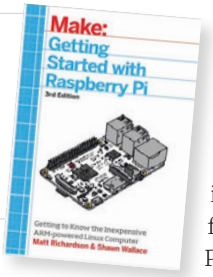


Authors: Ray Nicholus
Publisher: Apress
Price: £27.99
ISBN: 978-1484222348
magpi.cc/2f0x06R

The jQuery playpen is not just preventing you improving your coding, it's often an unnecessary dependence. Nicholus shows you a JavaScript-only way of working and lets you make your own decision.

MAKE:

GETTING STARTED WITH RASPBERRY PI



Author: Matt Richardson & Shawn Wallace
Publisher: Maker Media, Inc
Price: £14.50
ISBN: 978-1680452464
magpi.cc/2f0L1kN

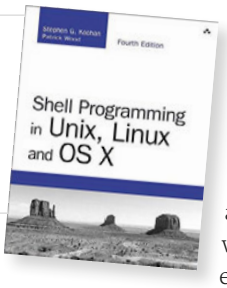
There are many newbie guides to the Raspberry Pi – unsurprisingly, given the numbers of Pis sold, and the diversity of users – and they range from the basic to the mind-bogglingly comprehensive. This title is in the former camp, being somewhat concise, and is in a pleasingly small format – but nothing essential is missing from this work.

After introducing the hardware, and booting up, we get a quick round-up of the command line. Each chapter finishes with a 'Going Further' section, and in this case, as well as some Linux references, the online Jargon File – which gives cultural background to several decades of ancestry behind Raspbian – is recommended. Thoughtful touches like this lift the book above many cobbled-together intros, as does the chapter selection, e.g. the brief round-up of interesting alternative OSes for the Pi, or the synergistic Pi/Arduino relationship.

Python chapters are necessarily brief, but the GPIO section gives enough working examples to get the reader started, as does the chapter on Python and the internet, which introduces Flask. The section on cameras is also very useful. This is a very handy little book that will carry newbies a long way.

Score ★★★★★

SHELL PROGRAMMING IN UNIX, LINUX AND OS X



Author: Stephen G. Kochan, & Patrick Wood
Publisher: Addison Wesley
Price: £27.99
ISBN: 978-0134496009
magpi.cc/2f0CCOw

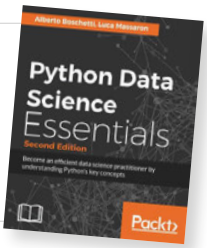
More than a quarter of a century after the first edition, this still-relevant classic gets a face-lift for today's developers. Though they are likely to be using the shell on a Mac or Linux box, rather than a Unix terminal, the book is applicable across all *nix-like environments. Kochan and Wood's successful teaching-by-example style is still present, but the book starts with a sensibly paced recap of basic commands, then the shell's relationship to the rest of the system, before giving you some tools to work with: wildcards, regular expressions (in an exemplary fuss-free intro to a subject that often causes confusion), cut and paste, sed, sort, and others.

Scripting time, and the authors don't disappoint as commands are introduced in a timely fashion, always with appropriate explanations of what's happening as they're executed. This continues as scripts build up, giving the reader an excellent understanding of how their operating system works behind the scenes.

The sample phone book program gets revisited later with consideration for data formatting and other lessons learned, then – after nearly 300 pages of pure, POSIX compliant shell scripting – a look at the additions offered by Bash (Raspbian's shell) and Korn (the basis for Zsh).

Score ★★★★★

PYTHON DATA SCIENCE ESSENTIALS



Author: Luca Massaron & Alberto Boschetti
Publisher: Packt
Price: £27.99
ISBN: 978-1786462138
magpi.cc/2f0HVxc

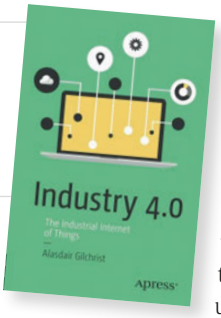
Python is dominating the complex field of data science, its libraries and ease of use enabling scientists to combine work in "linear algebra, statistical modelling, visualisation, computational linguistics, graph analysis, machine learning, business intelligence, and data storage and retrieval" to good effect. Prolific Python and data authors Massaron and Boschetti give us a work – expanded from the first edition – that should get any reasonably technical computer user started in understanding the field.

The book assumes basic Python knowledge, but presents routes to get you up to speed if necessary – the appendix on 'Strengthening your Python Foundations', with sections on code and places to learn (MOOCs, etc.). Each chapter, as well as teaching you the tools of data science – and Python's great libraries – will build on Python basics to give you a practical mastery of the language which could be useful in other fields.

From Jupyter Notebooks (formerly IPython), Numpy, Pandas, and Scikit-learn, through Theano and Tensorflow, to seaborn and ggplot visualisations, and web deployment with Bottle, the book retains its friendly walkthrough format throughout, including several practical examples. Data munging, machine learning algorithms, social network analysis, and more are all well covered, at an enjoyable pace.

Score ★★★★☆

INDUSTRY 4.0



Author: Alasdair Gilchrist
Publisher: Apress
Price: £19.99
ISBN: 978-1484220467
magpi.cc/2f0Hu68

In manufacturing, there is a synergy – that oft-abused word – of high-speed, low-cost communications and processing technologies with ever lower costs of storage and processing of data. Add in intelligent, self-monitoring sensors, and the ability to process and act upon mountains of data at real or near real-time speeds, and we're approaching the "fourth industrial revolution" detailed here by Alasdair Gilchrist.

This is a serious-minded look at the issue, somewhat dry at times, but very rewarding for anyone looking for a good round-up of the changes already observable in industry, and the technologies driving them. Indeed, many makers – particularly those moving into manufacture of their projects – could find something useful here.

After chapters on the industrial internet, and some of the network technologies driving the changes, the author moves up the stack to middleware then, after some of the interesting WAN technologies, software patterns and some of the security challenges. The value chain is next then, inter alia, the EU's IoT6 workgroup's promising results in replacing proprietary protocols, and use of augmented reality to prevent forklift truck accidents. After discussing POS changes, and smart factories, both present and future, there's a roadmap to the future.

Score ★★★★☆

ESSENTIAL READING: CHRISTMAS

'Tis the season of buying your loved ones useful and beautiful books to treasure.

The Smart Solution Book

Author: David Cotton
Publisher: FT Publishing International
Price: £16.99
ISBN: 978-1292142319
magpi.cc/2f0KjUV

Everybody's got problems, but Cotton has effective solutions to lift the gloom from a work-troubled colleague or relative.

Edible Inventions

Author: Kathy Ceceri
Publisher: Maker Media, Inc
Price: £14.50
ISBN: 978-1680452099
magpi.cc/2f0xs53

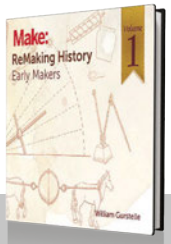
Food meets inventiveness: hydraulic 3D LEGO food printer, chemical cuisine, recipes, aquaponics, and a cardboard solar oven. Fun and inspirational.

Games, Design and Play

Author: Colleen Macklin & John Sharp
Publisher: Addison Wesley
Price: £30.99
ISBN: 978-0134392073
magpi.cc/2f0J1t5

Games design as an art form – producing games that people will love. Rules to take (and break) to any platform.

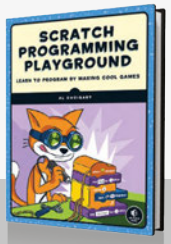
ReMaking History, Volume 1



Author: William Gurstelle
Publisher: Maker Media, Inc
Price: £14.50
ISBN: 978-1680450606
magpi.cc/2f0LJyv

A delightful and beautifully produced look at ancient inventors and inventions, and how to recreate the latter at home.

Scratch Programming Playground



Author: Al Sweigart
Publisher: No Starch
Price: £17.99
ISBN: 978-1593277628
magpi.cc/2f0G9Mv

From Breakout to Fruit Ninja, learning-as-fun for your 10- to 12-year-old – or for you!

SPY V. SPI

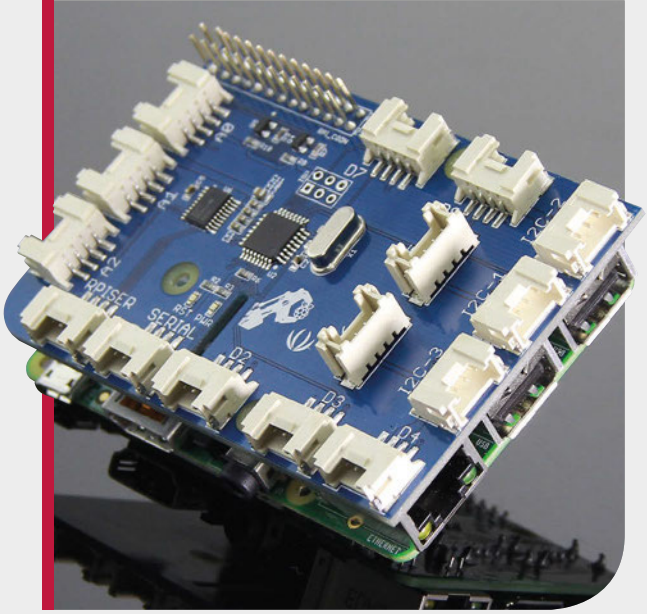
ROBOT CODING MISSIONS

CODE. CAPTURE. DEFEND

Donning disguises, we arrange a rendezvous with Dexter Industries' **Taryn Sullivan** to exchange information on a top-secret spy programme

EASIER ELECTRONICS

The GrovePi (**magpi.cc/2eEPX1U**) is an add-on board that plugs into the Raspberry Pi's GPIO pins, and features numerous sockets to plug in Grove sensors and other electronic components, of which over 100 types are available. This makes it much easier to wire up physical computing projects, without the need for a breadboard. The Base Kit, as included with the Spy v. sPi, features an ultrasonic distance sensor, dial sensor, button, light sensor, buzzer alert, sound sensor, power relay, temperature/humidity sensor, LCD screen, and red, green, and blue LEDs.



Shhhh - you didn't hear this from us... Dexter Industries is launching a fun educational game that teaches children to solve problems in the form of secret spy missions. To prevent their own 'jewel' from being stolen by rivals in this capture-the-flag-style contest, teams will build protective programmable devices using components from the supplied robot kit, ranging from distance, sound, light, and infrared sensors, to buttons, buzzers, and LEDs.

"We got the idea from some of our customers with kids that loved playing spy games," says Taryn Sullivan, COO at Dexter. "Then we met the guys from Building Momentum who develop and put on technical training of all kinds, and happen to know about the tech real spies use, and thought 'that's a perfect match!'"

While the Spy v. sPi project failed to hit the funding target in its recent Kickstarter campaign (**kck.st/2d5tyc6**), Dexter confirms it's definitely still going ahead. Missions come in the form of an illustrated colour booklet or online step-by-step instructions, including pictures, diagrams, extra resources, and videos. Each of the ten missions will teach players how to program the GrovePi board and an assortment of sensors to accomplish different objectives, all within a story created by 'real spies'. "Let's just say they were all in the intelligence field, and have set up their fair share of surveillance tools for recon missions," teases Taryn.

She tells us that missions will have a real-life activity to accomplish, such as learning how to set a tripwire, or detecting when someone's coming into your room. "Imagine you just hid your jewel in the perfect place and you want to 'bug' the door to the room, so you know when someone tries to enter. Spy v. sPi teaches you how to take the GrovePi and turn it into a real-life surveillance system. Using ultrasonic range finders and acoustic sensors, the sPis can invent unique and clandestine ways to monitor a door's movement or detect a sneaking opponent."

In the team game, opponents will sneak around and try to physically capture the other side's 'jewel', while defending their own using the robot kit. Quite what the jewel is remains a mystery at this point: "[It] will be revealed by the spy as they accomplish their missions. We can't give away too much!"

While there are step-by-step instructions to help players build their kits and code the sensors, each

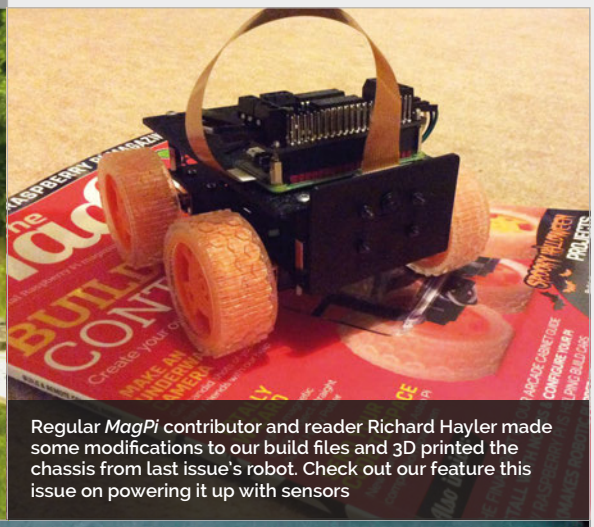
THE MONTH IN RASPBERRY PI

Everything else that happened this month in the world of Raspberry Pi

READER PHOTOS

You folks out there are the reason we write the magazine, and we love it when you send us photos of stuff you've done using the magazine. Last month, we had two physical projects of note in the mag: our tiny cover robot and the light-up pumpkin using GPIO Zero code. People have already started making the robot, and one young reader made the pumpkin for Halloween himself! We also got a wonderful picture from a Naturebytes wildlife cam, which you can see on the left…

The students of Fleetville Junior School in St Albans caught this candid photo of a muntjac deer with their Naturebytes camera. Since putting it up, they've caught foxes and squirrels milling about



Regular *MagPi* contributor and reader Richard Hayler made some modifications to our build files and 3D printed the chassis from last issue's robot. Check out our feature this issue on powering it up with sensors

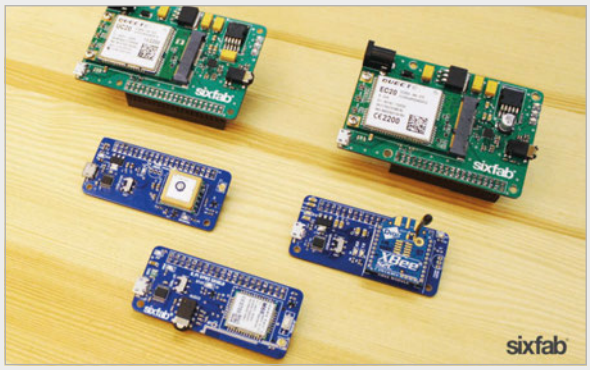
11-year-old Louis went above and beyond at Halloween and made this awesome light-up pumpkin using our tutorial from last issue. He especially liked how the LED had a candle flickering effect to it

SEND US YOUR CHRISTMAS TREE COVER PHOTOS!

magpi@raspberrypi.org

CROWDFUND THIS!

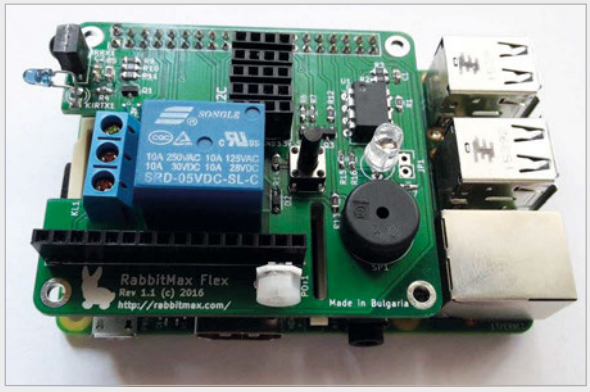
The best crowdfunding hits this month for you to check out...



RASPBERRY PI IOT SHIELD FAMILY

kck.st/2evMGys

Instead of being an all-in-one HAT, this is a modular family of different IoT devices that you can build up on your Pi or Pi Zero for different functionality. GPRS, 3G, and 4G/LTE shields give you cellular internet access (with a SIM inserted) so you can use your Pi on the go. There are also GPS and XBee radio shields. The Kickstarter campaign has a while to run, so you should have plenty of time after reading this to decide on it.



RABBITMAX FLEX

magpi.cc/2eFC8gr

The RabbitMax Flex (excellent name) is an IoT HAT for the Raspberry Pi that doesn't require any soldering. Not all HATs do require soldering, but it's nice to know when one doesn't in case you're slightly anxious about picking up the iron. This board comes with a number of features, such as IR, a buzzer, a button, an RGB LED, and slots for standard extra sensors. Check it out on its Indiegogo page.

PETITION MOJANG TO UPDATE MINECRAFT PI

magpi.cc/2eFyTpv

Instead of money, this month we want to crowdsource your opinion and votes! Minecraft Pi on the Raspberry Pi is excellent, but it could use an update and/or continual development. With that in mind, a new petition has been started on Change.org to try to convince Mojang, its developer, to resume work on it. Give it a sign and maybe we can make it happen.

BEST OF THE REST

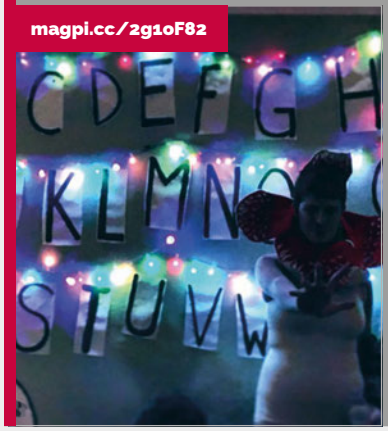
Here are some other great things we saw this month

magpi.cc/2g1ot80

DELOREAN CASE

Where we're going, we don't need *Back to the Future* quotes. Reddit user aldog3788 took this model of the *BttF* time-travelling DeLorean and installed a Raspberry Pi into it. When this baby hits 88 seconds after boot, you're going to see some serious Scratch.

magpi.cc/2g1oF82



STRANGER THINGS WALL

Possibly a little late to make your own for Halloween, but here's a Pi-powered alphabet wall. It's interactive, accepting text messages and even being able to give you yes or no answers to questions. Maybe a good use for your Christmas lights before you put them away in January?

mission has room for every spy to come up with their own unique way to complete it. In addition, there's the option of playing alone: "Solo missions enable spies to go on their own missions, without needing an opponent. They will also learn to defend their room and use their spy device around their house."

Modular components

To make things easier for newcomers, the hardware kit includes Dexter's GrovePi board which enables the Raspberry Pi (not included) to be easily connected to Grove modular sensors, rather than having to wire up standard electronic components. "The GrovePi makes it easy to get started with robotics because it has a board that sits on the Raspberry Pi, and comes with a collection of plug-and-play modules and sensors. While breadboarding can be fun, the GrovePi kit allows you to get to programming your device easier and faster."

When it comes to coding, missions are available in both Scratch and Python versions, so you can choose which one you'd like to use. "We found that while some people grow out of Scratch to Python, others are doing super-complicated things with Scratch and don't need to. Likewise, some prefer text-based languages like Python, so we wanted to offer that option too."

Making it accessible for everyone is a key aim for Dexter, which responded to some early feedback from testers: "At first they thought it was too hard, so we went back and revised the whole set of missions in a way that makes it really accessible for newbies, but kept the more advanced challenges in there for the experts."

> "Solo missions enable spies to go on their own missions, without needing an opponent"

The gaming aspect was also a major factor. "One of the reasons we decided to include a competitive aspect to this new programme is because we had so much interest from schools, camps, and after-school programs for GoBox," says Taryn. "GoBox is more of an individual project based on the GoPiGo, a robot car. But we heard from educators that students are often encouraged by competition, so we thought this was a perfect narrative to allow for that kind of activity."

While ten missions have been created so far, more may be added later. "We do envision offering more missions for spies to unlock as they advance, perhaps even using our new content platform, Dexter Studios, to be an avenue for spies to share missions with each other, or challenge spies to accomplish what they have come up with."

GOBOX ROBOTICS COURSE

Dexter's previous educational programme, GoBox – the world's first subscription-based robot kit (**magpi.cc/2fp2pAg**) – was launched at the end of last year, as reported in *The MagPi* issue 32. Designed for newcomers to robotics and programming, the year-long course's monthly modules see users building a GoPiGo robot car and equipping it with various sensors to complete missions.

COMMUNITY PROFILE

ALEX EAMES
RASPI.TV

When he's not helping to expand the Raspberry Pi community, Alex is a keen macrophotographer and knife maker!

Alex Eames

Category: Maker, teacher, and YouTuber

Day job: Full-time RasPi.TV guy

Website: raspi.tv

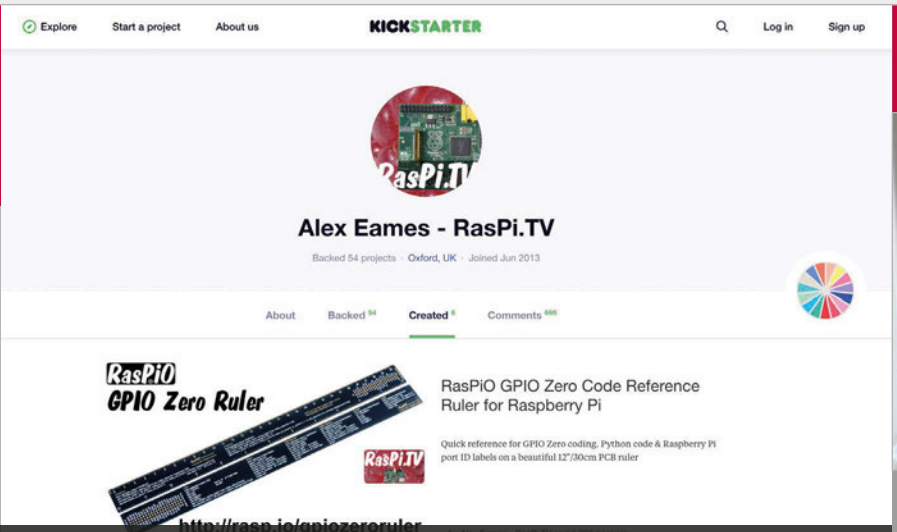
Alex purchased his first Raspberry Pi in May 2012 after a BBC article caught his eye. Already teaching ICT at his son's school, he was drawn to the idea of a $35 computer to aid the education of his ten-year-old students.

That same month, Alex started his website, RasPi.TV, which allowed him both to document his progress with the Raspberry Pi and to curate an easy-to-use reference library for others.

"I found that when I wanted to learn something new, generally the 'instructions' on other Linux sites were either out of date or incomplete. I wanted a place where I could record procedures that I could use again, but that would also be available to others."

Alex was determined to provide tutorials that worked first time, understanding the frustration of newcomers when their hard work didn't always pay off. "It's off-putting for people to follow a list

Below Alex is truly a member of the Raspberry Pi community, providing support and resources both to those new to, and experienced in, the world of the Pi



The 'Kickstarter King' has continued to bring his builds to the crownfunding network, building the RasPiO family in the process (magpi.cc/2eF7ML9)

of instructions, get it all right, and then find the process fails," he says. RasPi.TV was all about "instructions that work first time – even if you've never done it before."

In 2012, Alex began to build his own RasPiO boards, with the first releases making an appearance in March 2014. The GPIO labeller, Breakout, and Breakout Pro were successful across the community, earning an honourable mention on the official Raspberry Pi blog. The Pro has since been upgraded to the Pro HAT, while the labeller has been replaced with a newer 40-pin version. The RasPiO collection has now increased to ten different units, each available for direct purchase from the website. A few originally found their feet via successful crowdfunding campaigns.

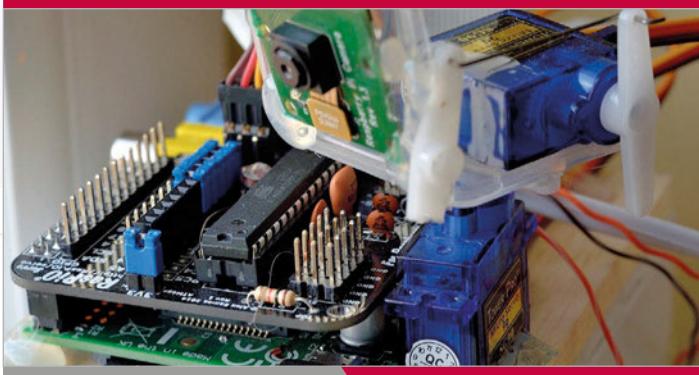
> "I wanted a place where I could record procedures to use again, also available to others"

Even if you've yet to visit either RasPi.TV or Alex's YouTube channel, the chances are that you've seen one aspect of his online contribution to the Raspberry Pi Community. Alex maintains a Raspberry Pi 'family photo' on his website (**raspi.tv/rpifamily**), showcasing every model built across the years. It's a picture which often does the rounds of blogs, news articles, and social media.

Outside of his life of Pi, Alex has a background in analytical chemistry, a profession that certainly explains his desire for the clean, precise, and well-tested tutorials that brought about the creation of RasPi.TV. From working as a translator to writing his own e-books, Alex is definitely well suited to the maker life, moving on from his past life of pharmaceutical development. His tutorial and review videos on YouTube reach viewing figures in the thousands, with his popular Raspberry Pi DSI Display Launch video garnering close to 300,000 views at the time of writing of this article.

While Alex has updated us on his newest unreleased projects and plans, we'll keep them quiet for now. You'll have to watch the RasPi.TV website for details.

HIGHLIGHTS



DUINOCAM SECURITY CAMERA

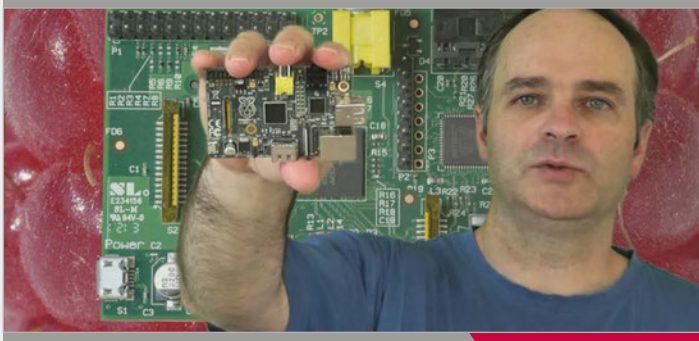
magpi.cc/2eF6uzO

The Duinocam is set up in Alex's home in Poland. During daylight hours, it emails him photos and temperature data while also responding to tweeted commands such as video capture and upload. Using a Pi Model B, a RasPiO Duino, a Camera Module, and two servos, the unit can pan and tilt to survey the area.

RASPIO

rasp.io/introducing-raspio

The RasPiO family is a series of add-on boards, port labellers, GPIO rulers, and tools to aid makers in building with the Raspberry Pi. The ruler, for example, offers GPIO pin reference for easy identification, along with code reference for using the GPIO Zero library.



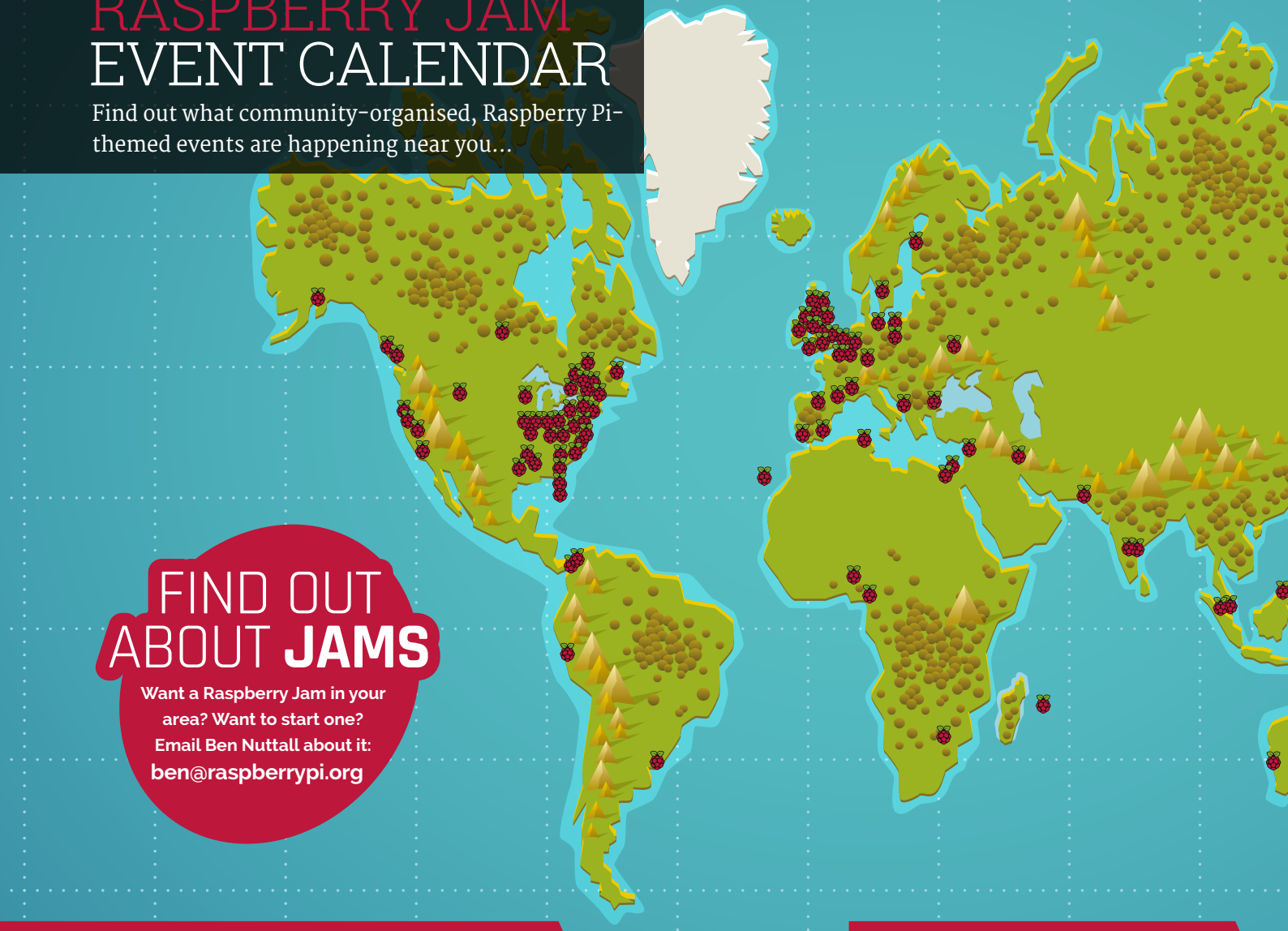
RASPI.TV

raspi.tv

The RasPi.TV website is packed full of tutorials, reviews, and videos, all of which have the aim of helping both newcomers and seasoned Raspberry Pi users to expand their skill set and interests. Alex's YouTube channel boasts over 8,000 subscribers, with viewing figures of well over 1.5 million across his 121 videos.

RASPBERRY JAM EVENT CALENDAR

Find out what community-organised, Raspberry Pi-themed events are happening near you...



FIND OUT ABOUT **JAMS**

Want a Raspberry Jam in your area? Want to start one? Email Ben Nuttall about it:
ben@raspberrypi.org

HIGHLIGHTED EVENTS

1 SITTINGBOURNE RASPBERRY JAM

When: Saturday 10 December
Where: No. 34, Sittingbourne, UK
magpi.cc/2g0InRj
An interactive event that lets you get to grips with the Raspberry Pi and its software.

2 RASPBERRY JAM HORSHAM

When: Sunday 11 December
Where: Parkside, Horsham, UK
magpi.cc/2g0Lk45
Come and have fun exploring, learning, and using the Raspberry Pi for coding, robotics, and electronics.

3 STAFFORD RASPBERRY JAM

When: Tuesday 13 December
Where: The Signpost Centre, Stafford, UK
magpi.cc/2g0GT9x
A big meetup of Raspberry Pi enthusiasts to share ideas, help each other, and have fun.

4 EAGLE LABS RASPBERRY JAM #2

When: Saturday 7 January
Where: Serendip, Birmingham, UK
magpi.cc/2g0LAjs
There'll be a range of things to explore, such as Pi projects, 3D printing, Minecraft coding, and more!

REGULAR EVENTS

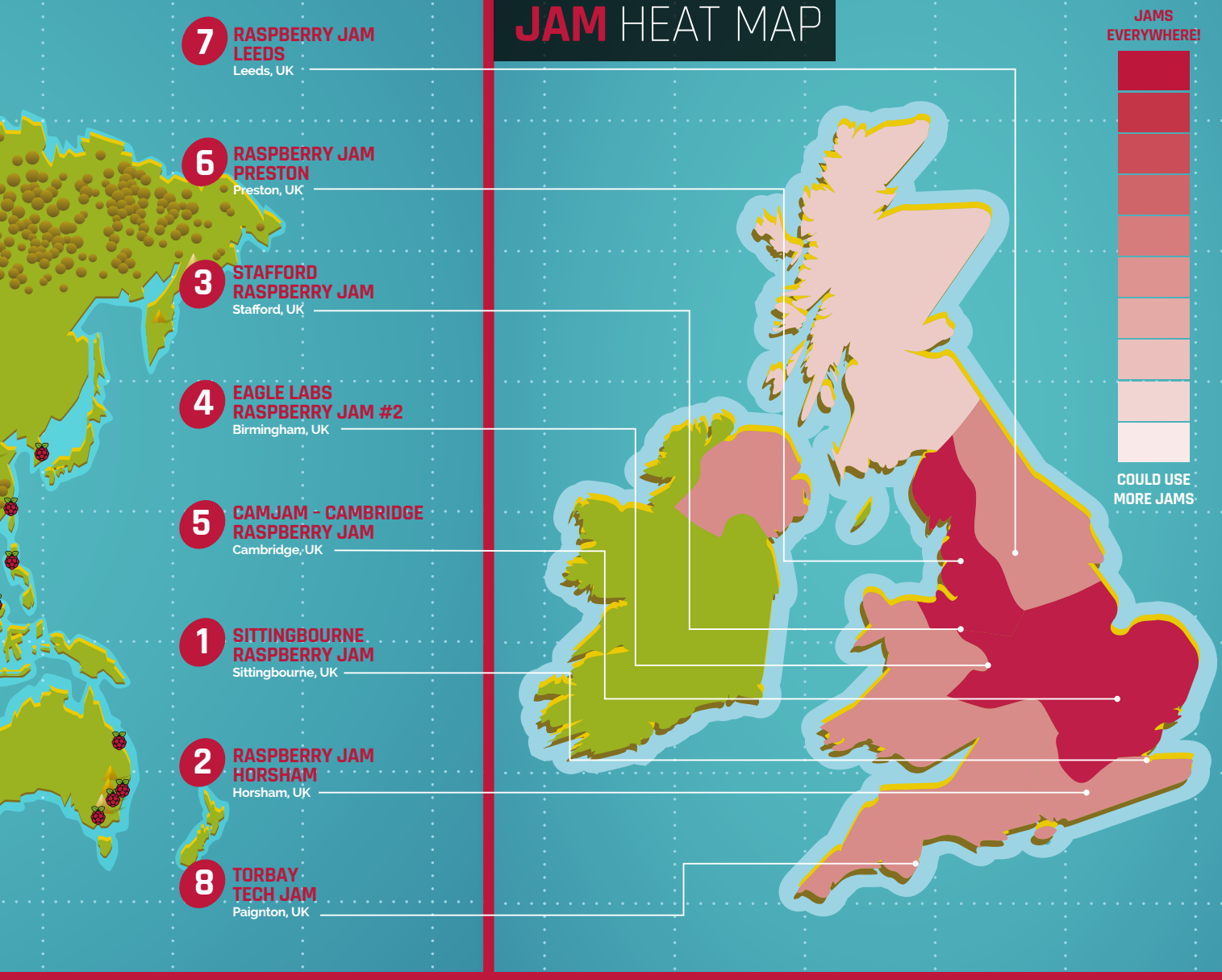
5 CAMJAM - CAMBRIDGE RASPBERRY JAM

When: Saturday 3 December
Where: Institute of Astronomy, Cambridge, UK
magpi.cc/2eF2a3k
The hometown event is back again this December to see out the year with one last Jam.

6 RASPBERRY JAM PRESTON

When: Monday 5 December
Where: Media Innovation Studio, Preston, UK
magpi.cc/2g0EoE6
Learn, create, and share the potential of the Raspberry Pi at a family-friendly event.

JAM HEAT MAP



7 RASPBERRY JAM LEEDS

When: Wednesday 7 December
Where: Swallow Hill Community College, Leeds, UK
magpi.cc/2g0KVi5
Everyone is invited for a couple of hours of computing fun, talks, demonstrations, and hands-on workshops.

8 TORBAY TECH JAM

When: Saturday 10 December
Where: Paignton Library and Information Centre, Paignton, UK
torbaytechjam.org.uk
A fun, informal, and family-friendly event that aims to inspire people to get into code and take up STEM subjects.

FILL IN THE GAPS!

IRELAND NEEDS RASPBERRY JAMS

Another region that Ben Nuttall, Raspberry Jam community manager, has indicated is in need of Raspberry Jams is Ireland. There have been Jams in Dublin, along with a regular one in Belfast hosted by Andrew Mulholland. However, it would be great to get more of them through both Northern Ireland and the Republic of Ireland. Check out our previous Jam interviews for tips on how to start a Jam, and try to bring a bit of computing education to your local community!

YOUR LETTERS

Three generations

The Raspberry Pi computer has been a really great asset for me.

Let me first give you my story. I'm no great computer whiz, although I have been around computers since the days of DOS (when I was knee-high). I've seen from Windows 3.1 all the way through to Windows 10, and I've played around with various Linux OSes, but I never really got involved with the coding side. I never knew coding.

Now enter the Raspberry Pi. My first Pi was a Model 1 B, which ultimately became a Kodi box. When I upgraded to the B+, I gave the original Raspberry Pi to my brother who is still using it as a Kodi box. I gave the next to my Dad and showed him how to use it as a Kodi box too. Then I got two Pi 3s, one of which I'm using as a Kodi box (do you see a trend?), and, after much trial and error, I've set the other up as a Samba media server.

This is what taught me a lot about coding, and although my knowledge is far from complete, far from even scratching the surface, it's given me the confidence to want to attempt my next project: an internet radio. One Pi I've set up as my son's desktop, I introduced him to Scratch, and we haven't seen him since... I think he's still there playing around on it.

We're just going to sneakily plug our new range of Essentials books that are now in print here, including a Scratch book

So that is three generations of my family using it with confidence, with me planning on getting my sixth Pi. I cannot wait to see what the Raspberry Pi Foundation and Eben Upton bring us next (when is the Pi 4 coming out?).

It's a great little computer with loads of potential most of us never actually unlock. I've always read *The MagPi* and cannot wait for the next issue to be available.

A little help

I have a project I'm working on that operates without a keyboard, mouse, and display. The OS which I will be using to power the Raspberry Pi is Raspbian. So it needs to boot up and not ask for a password, then it should start a Python 3 script. Any suggestions?
Michael Mouton

You're in luck, Michael. This is actually quite simple to do (once you know how). We talked about it in the cover feature, but if you missed it there, here's what you need to do...

First of all, Raspbian by default will automatically log in after booting up. However, if you're not using it to display anything, we'd suggest making it boot to command line. This way, it turns on faster and will use a little less power at the same time. From the top-left menu in the Raspbian desktop, select Preferences and then Raspberry Pi Configuration. From there you can change the option from boot to desktop to boot to CLI. Keep the automatic login option the same and when you next reboot, it will automatically load into the command line.

To have a program run after it boots, make sure it has `#!/usr/bin/env python` on the first line of the script. Save it to the **pi** home folder, then open up the profile file with `sudo nano /etc/profile` and add the line `sudo python [filename].py` to the end of the file.

Thank you to all those involved for just being awesome.
Terence

Thanks for your email, Terence. It's great to hear how so many people in your family are using the Pi for something, even if it is just a bit of Kodi! If your son is really getting into Scratch, make sure to show him the *Scratch Essentials* book that we have out right now – it should help him learn a thing or two and keep him out of sight for a bit longer!

As for the Raspberry Pi 4, you may have to wait a little while for that, but we'll let Eben know you're looking forward to it.

FROM THE FORUM:

The Raspberry Pi Forum is a hotbed of conversations and problem-solving for the community – join in via **raspberrypi.org/forums**

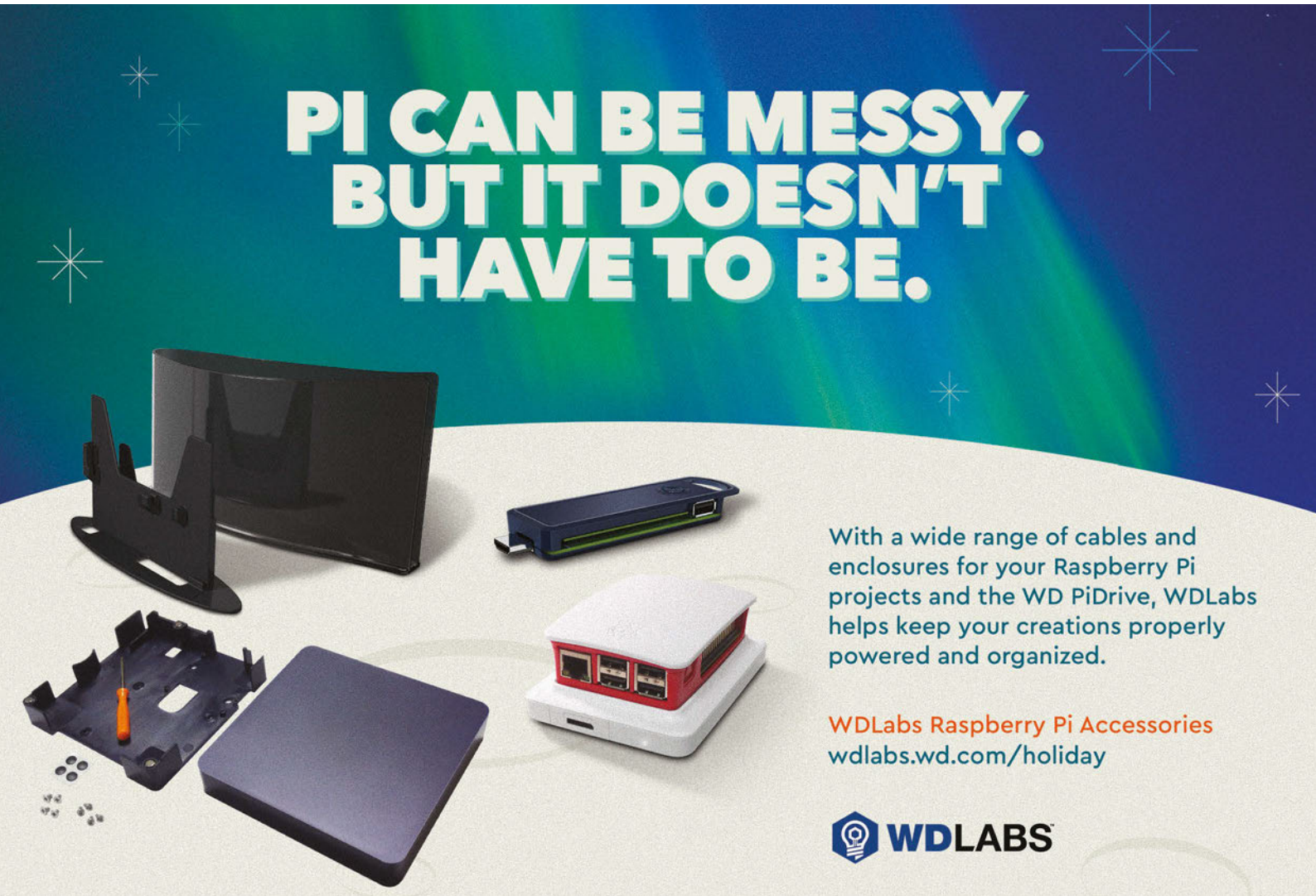
KEY ISSUES

Reader Ian Williams asks about the keyboard in the official starter kits – he wants to know whether English orders will contain a UK keyboard because the position of **"** and **@** is reversed, which can cause problems when coding if using the US keyboard. This is Linux country… I assume changing the code from a keypress to another character is easy. There are plenty of instruction sites online. The editor of the page seems to only suggest rewiring the brain or buying a cheap keyboard from elsewhere. Is the Pi somehow different, which makes key changes impossible?
Ken

Hi Ken. Ian was asking specifically about the physical layout of the keyboard, which is currently a US one. You're right, though: the input configuration can be changed to work like a UK keyboard in Raspbian. From the top-left menu, select Preferences, then Raspberry Pi Configuration. Click the Localisation tab, then Set Keyboard and you'll be able to choose the layout.

WRITE TO US

Have you got something you'd like to say?
Get in touch via **magpi@raspberrypi.org** or on The MagPi section of the forum at: **raspberrypi.org/forums**

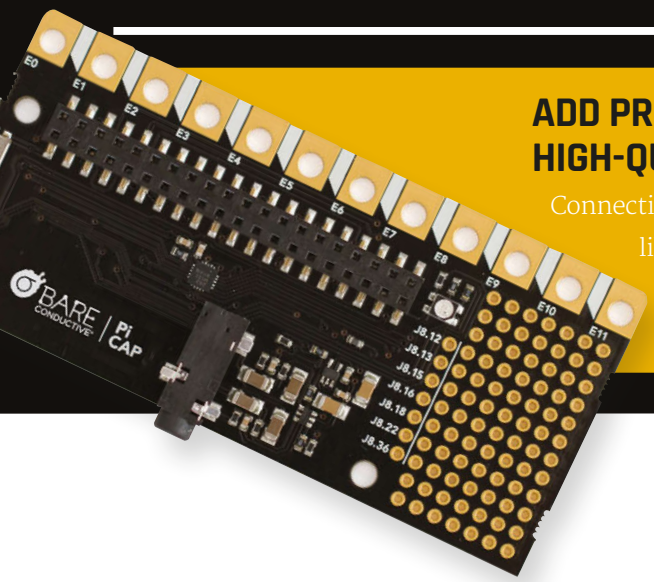

PI CAN BE MESSY.
BUT IT DOESN'T
HAVE TO BE.
With a wide range of cables and enclosures for your Raspberry Pi projects and the WD PiDrive, WDLabs helps keep your creations properly powered and organized.
WDLabs Raspberry Pi Accessories
wdlabs.wd.com/holiday
WDLABS

In association with

BARE CONDUCTIVE®

bareconductive.com

WIN 1 OF 5 PI CAPS EACH WITH A 10 ML TUBE OF ELECTRIC PAINT!



ADD PRECISE CAPACITIVE TOUCH, DISTANCE SENSING, AND HIGH-QUALITY AUDIO TO YOUR RASPBERRY PI WITH PI CAP.

Connecting the physical world to the digital world is easy with our fully featured libraries and extensive online tutorials. You can also use Pi Cap with Electric Paint: nontoxic, solvent free, and water soluble, you can use this to paint wires or sensors directly onto almost any material.

WHAT'S **DIFFERENT** ABOUT BARE CONDUCTIVE'S **SPECIAL PAINT?**

Tell us by 19 December for your chance to win!

Simply email **competition@raspberrypi.org** with your name, address, and answer!

Terms & Conditions

Competition closes on 19 December 2016. Prize is offered worldwide to participants aged 18 or over, except employees of the Raspberry Pi Foundation, the prize supplier, their families or friends. Winners will be notified by email after the draw date. By entering the competition, the winner consents to any publicity generated from the competition, in print and online. Participants agree to receive occasional newsletters from The MagPi magazine (unless otherwise stated upon entry). We don't like spam. Participants' details will remain strictly confidential and won't be shared with third parties. Prizes are non-negotiable and no cash alternative will be offered.

Love your Pi? Love Music?

Designed by audio experts, enjoyed by everyone

Made in the UK

10% Discount voucher "MagPi10"

Jaw dropping audio quality for your Raspberry Pi

Connect Headphones, Speakers, RCA, Toslink, S/PDIF or XLR

We work with all the major music solutions to ensure compatibility

Sonic Pi · moOde audio player · OSMC · piCorePlayer SQUEEZE AUDIO · VOLUMIO · roon · Archphile · PiMusicBox · KODI · rune audio · MAX2PLAY

Twitter: @IQ_audio
Email: info@iqaudio.com
Web: www.iqaudio.com

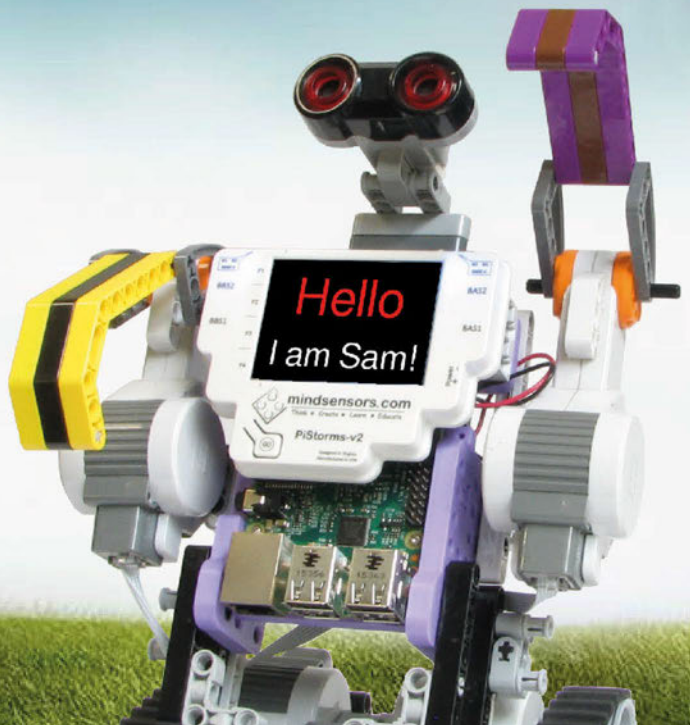
IQaudIO

IQaudio Limited,
Cricklade Wiltshire.
Company No.: 9461908

mindsensors.com

Think • Create • Learn • Educate

!! Special !!
15% discount:
Use Code:
MAGPISPECIAL



PiStorms-v2

Make Stunning Robots with LEGOs and Raspberry Pi !

MATT RICHARDSON

Matt is Raspberry Pi's US-based product evangelist. Before that, he was co-author of *Getting Started with Raspberry Pi* and a contributing editor at *Make:* magazine.

HOLIDAYS WITH PI

For **Matt Richardson**, the holidays can be a time for digital making with family

When I was a kid, it felt like it took forever for the holidays to arrive. Now that I'm an adult, the opposite is true. It feels like the holidays come hurtling at us faster and faster every year. As a kid, I was most interested in opening presents and eating all of that amazing holiday food. As an adult, I mostly enjoy the opportunity to put a pause on real life for a few days and spend time with my family. Though I do still love eating all that amazing holiday food!

Invariably, the conversations with my extended family turn to Raspberry Pi at some point during the holidays. My relatives may have seen something in the news about it, or perhaps they have a friend who is creating their own retro gaming emulator with it, for example. I sometimes show off the Raspberry Pi projects that I've been working on and talk about what the Raspberry Pi Foundation is doing in order to put the power of digital making into the hands of people around the globe.

All over the world there will be a lot of folks – both young and old – who may be receiving Raspberry Pis as gifts during the holidays. For them, hopefully it's the start of a very rewarding journey making awesome stuff and learning about the power of computers.

The side effect of so many people receiving Raspberry Pis as gifts is that around this time of year we get a lot of people asking, "So I have a Raspberry Pi… now what?" Of course, beyond using it as a typical computer, I encourage anyone with a Raspberry Pi to make something with it. There's no better way to learn about computing than to create something.

There's no shortage of project inspiration out there. You'll find projects that you can make in this very issue of *The MagPi*, as well as all of the back issues online, which are all available as free PDF files. We share the best projects we've seen on our blog, and our resources section contains fantastic how-to projects from our education team.

Be inspired

You can also explore sites such as Hackster.io, Instructables, Hackaday.io, and Makezine.com for tons of ideas for what you can make with your Raspberry Pi. Many projects include full step-by-step guides as well. Whatever you're interested in – from music, gaming, electronics, natural sciences, to aviation – there's sure to be something made with Raspberry Pi that'll spark your interest.

If you're looking for something to make to celebrate the holiday season, you're definitely covered. We have seen so many great holiday-related Raspberry Pi projects over the years, such as digital advent calendars, Christmas light displays, tree ornaments, digital menorahs, and New Year countdown clocks. And of course, not only does this issue of *The MagPi* contain a few holiday-themed Pi projects, you can even make something festive with the cover and a few LEDs.

There's a lot of stuff out there to make and I encourage you to work together with your family members on a project, even if it doesn't seem to be their kind of thing. I think people are often surprised at how easy and fun it can be. And if you do make something together, please share some photos with us!

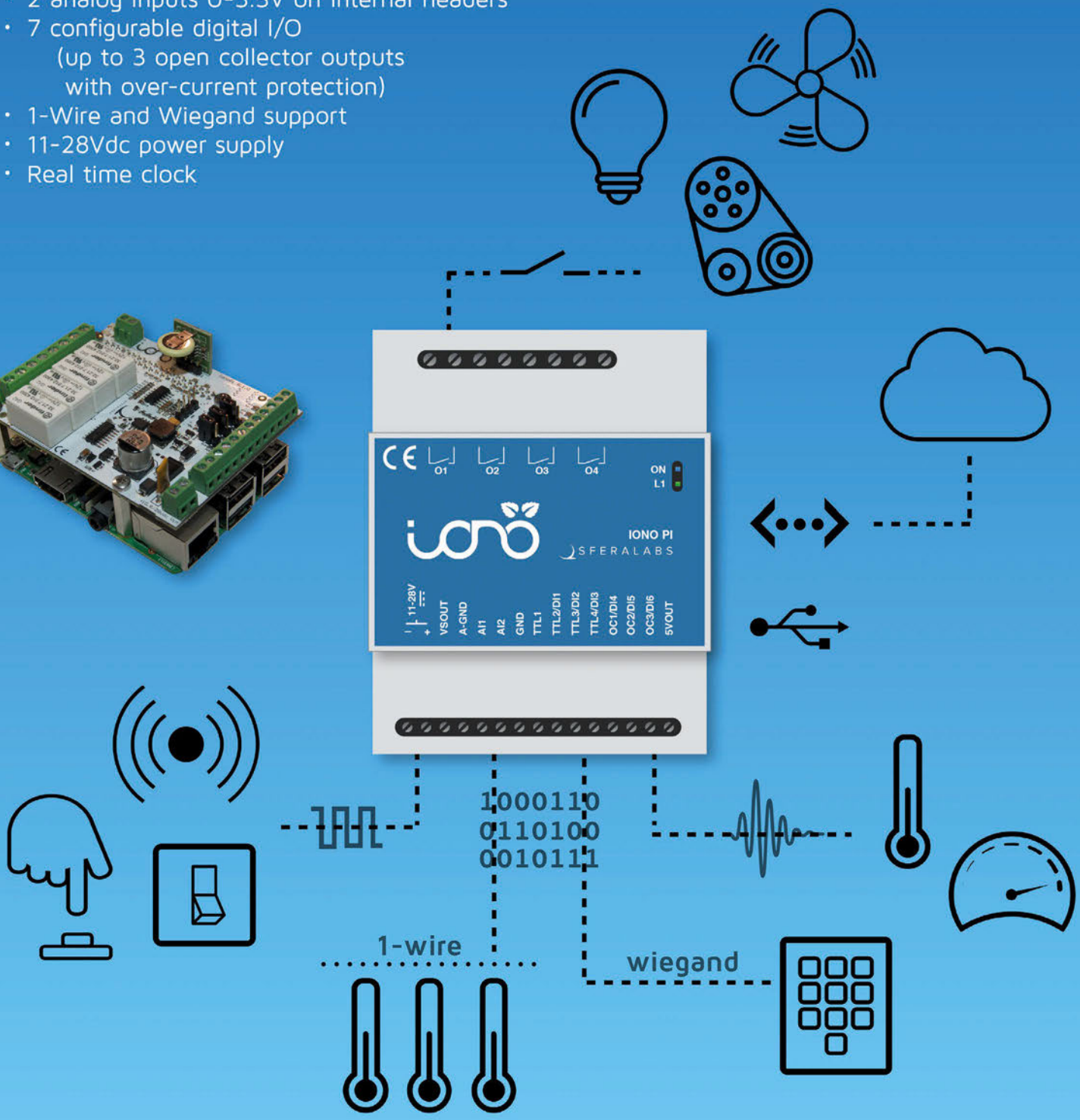
Whatever you create and whatever holidays you celebrate, all of us at Raspberry Pi send you our very best wishes of the season and we look forward to another year ahead of learning, making, sharing, and having fun with computers.

Iono Pi

A professional I/O expansion
for your Raspberry Pi

Board or pre-assembled DIN module with:

- 4 relay outputs 250V - 6A
- 2 analog inputs 0-30V
- 2 analog inputs 0-3.3V on internal headers
- 7 configurable digital I/O
  (up to 3 open collector outputs
  with over-current protection)
- 1-Wire and Wiegand support
- 11-28Vdc power supply
- Real time clock



SFERALABS

www.sferalabs.cc

THE NEW SB COMPONENTS ROBOT CHASSIS RANGE

SB Components

CREATE YOUR OWN
REMOTE CONTROLLED CAR
OR **OBSTACLE-SENSING ROBOT**

The perfect companion for the SB Components' **Motor-Shield**

FEATURES INCLUDE:

- Light-weight aluminium
- Electric motors
- Large wheels
- Stackable chassis plates

www.**robots.sb-components**.co.uk Call: **0203 514 0914**

At **SB Components** we strive to offer our customers the best prices for the best products. Our product team works tirelessly to source top quality affordable components from around the world.

Raspberry Pi is a trademark of the Raspberry Pi Foundation. Raspberry Pi not included.

The MagPi ESSENTIALS

LEARN | CODE | MAKE

OUT NOW IN PRINT

ONLY **£3.99**

from

magpi.cc/Back-issues

From the makers of the official Raspberry Pi magazine

GET THEM DIGITALLY:

Available on the App Store

GET IT ON Google Play

Expand your Pi

Stackable expansion boards for the Raspberry Pi

Serial Pi Plus

RS232 serial communication board. Control your Raspberry Pi over RS232 or connect to external serial accessories.

Breakout Pi Plus

The Breakout Pi Plus is a useful and versatile prototyping expansion board for the Raspberry Pi

ADC Differential Pi

8 channel 18 bit analogue to digital converter. I²C address selection allows you to add up to 32 analogue inputs to your Raspberry Pi.

IO Pi Plus

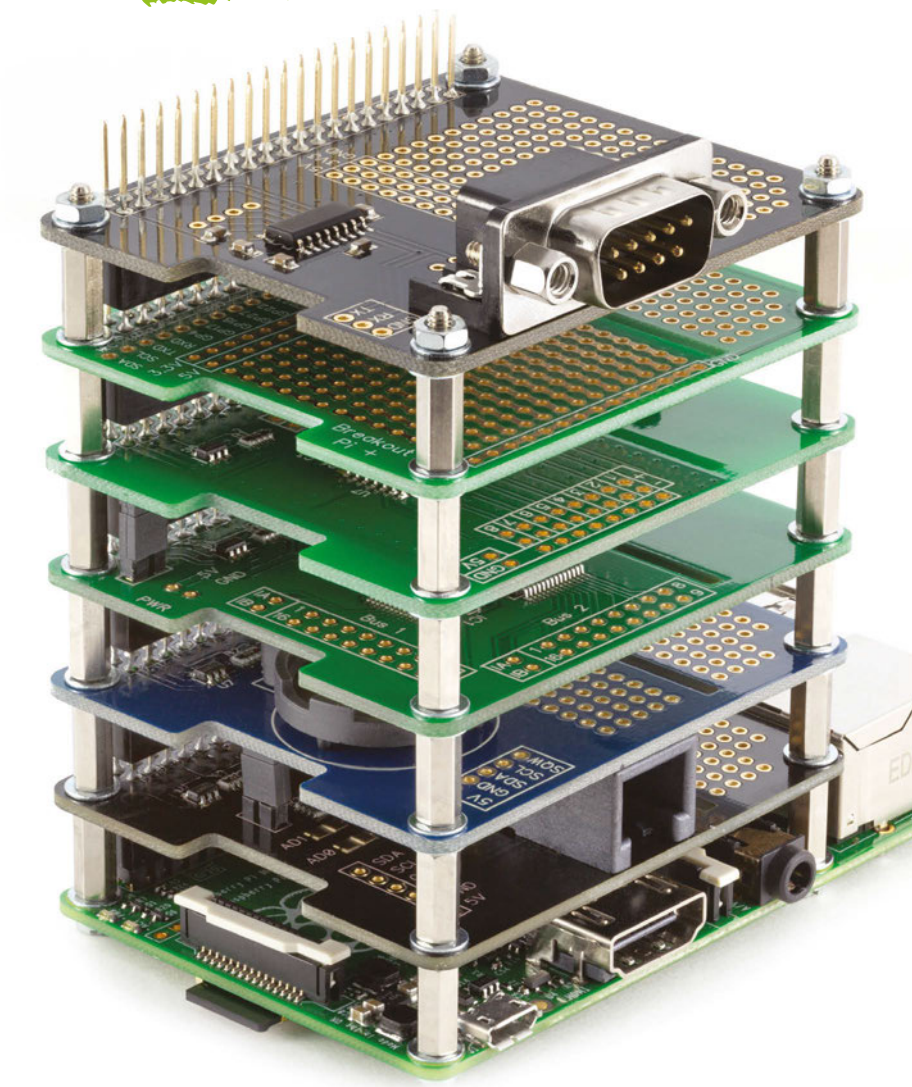
32 digital 5V inputs or outputs. I²C address selection allows you to stack up to 4 IO Pi Plus boards on your Raspberry Pi giving you 128 digital inputs or outputs.

RTC Pi Plus

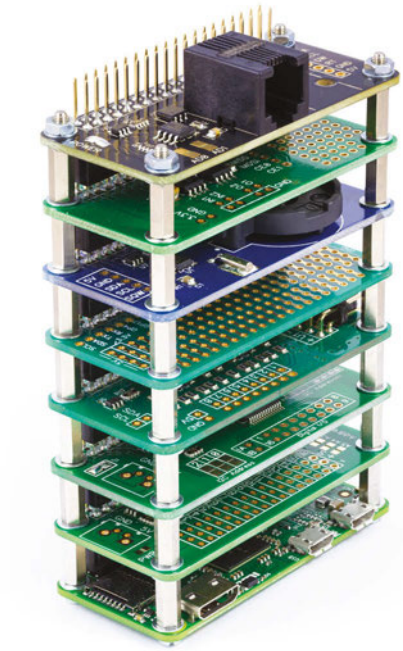
Real-time clock with battery backup and 5V I²C level converter for adding external 5V I²C devices to your Raspberry Pi.

1 Wire Pi Plus

1-Wire® to I²C host interface with ESD protection diode and I²C address selection.



Also available for the **Pi Zero**



ABelectronics UK

www.abelectronics.co.uk

DEXTER INDUSTRIES

SAVE 15%
coupon code
magpi

GrovePi
Plug-and-play Sensor Kit for Raspberry Pi
$89.99 USD-$179.99 USD

GoPiGo
A Robot Car with the Raspberry Pi
$99.99 USD-$199.99 USD

BrickPi
Raspberry Pi + LEGO MINDSTORMS
$99.99 USD-$199.99 USD

www.dexterindustries.com